CH568 Datasheet

Overview

CH568 is a high-performance 32-bit RISC reduced instruction set microcontroller, built-in 192KB FLASH flash memory, 32KB SRAM and 32KB DataFlash. On-chip integrated high-speed USB2.0 master/slave controller, 4 groups of SD controllers, SATA controllers, encryption algorithm modules, 4 groups of UARTs, 7 groups of PWMs, 3 groups of timers and other rich peripheral resources, which can be widely used for various embedded applications.

Features

- **Core:**
  - 32-bit RISC reduced instruction set core
  - 120MHz maximum frequency
- **Memory:**
  -192KB bytes program memory, support write protection
  - 32KB bytes SRAM
  - 32KB bytes DataFlash
- **USB2.0 high-speed receiver-transmitter (built-in PHY):**
  - High-speed Host / Device mode
  - Support control/bulk/interrupt/synchronous transmission
  - Support double buffering PINT-PONG
  - Support DMA
- **4 groups of independent SD controllers:**
  - Support single-wire, 4-wire, 8-wire communication mode
  - Support SD/TF card, SDIO card and eMMC card, etc.
  - Built-in FIFO
  - Support AES and SM4 Algorithms
  - Provide 8 encryption and decryption modes
  - Support DMA
- **SATA controller (built-in PHY):**
  - Support 1.5G/3G mode
  - Support power management
  - Support automatic data flow control
  - Support DMA
- **Timer:**
  - 3 sets of 26-bit timers
  - Support signal width sample/edge capture, PWM adjustable output, count function
  - TMR1 and TMR2 support DMA

  PWMX:
  - Expand 4 sets of PWM output
  - Adjustable duty cycle
- **Universal asynchronous receiver/transmitter (UART):**
  - 4 groups of independent UARTs, compatible with 16C550
  - The highest baud rate is 6Mbps
  - Built-in FIFO, multiple trigger levels
- **SPI:**
  - 2 sets of SPIs: One supports Master and Slave mode, the other supports only Master mode
  - Built-in FIFO
  - SPI0 supports DMA
- **LED screen interface:**
  - Support 1/2/4 channel data line
  - Built-in FIFO, support double buffering
  - Support DMA
- **Low power:**
  - Sleep mode
  -Support some GPIO, USB or SATA signal wake-up
- **General-purpose I/O port:**
  - 26 GPIOs
  - 8 pins can be configured level or edge interrupt
  - Some pins have multiplexing and mapping functions
- **ID Number of chip:**
  - Unique 64bit ID identification number
- **Power:**
  - 3.0~3.6V(3.3V±10%)
- **Package: LQFP48**

Applications

Security storage, home security, USB-related applications, monitoring, alarm systems, printers, scanners and other application control.

Chapter 1 Pin Information

1.1 Pin Configuration

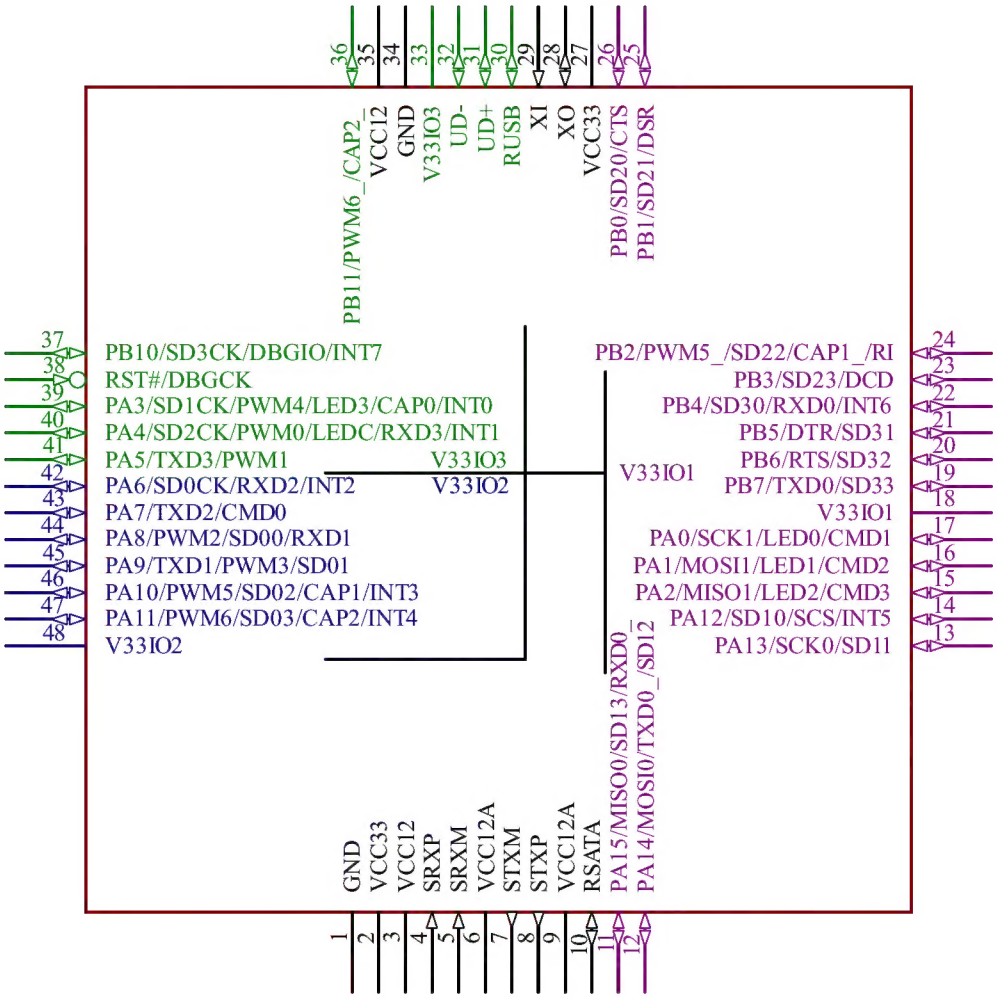


Figure 2-1 Pin arrangement of LQFP48 package

1.2 Pin Description

Pin No.	Pin Name	Type	Main Function (after reset)/ Multiplexing Function and Mapping	Function Description
1	GND	P	GND	Ground: common ground, 0V reference point.
2	VCC33	P	VCC33	The power voltage regulator inputs 3.3V power, which requires an external 0.1uF capacitor.
3	VCC12	P	VCC12	Core power output, and an external 3.3uF or 10uF capacitor is required.
4	SRXP	SATA	SRXP	SATA data receives positive polarity signals.
5	SRXM	SATA	SRXM	SATA data receives negative polarity signals.
6	VCC12A	P	VCC12A	SATA-PHY power, requires an external 1.2V

				power and an external 0.1uF capacitor.
7	STXM	SATA	STXM	SATA data sends negative polarity signals.
8	STXP	SATA	STXP	SATA data sends positive polarity signals.
9	VCC12A	P	VCC12A	SATA-PHY power, requires an external 1.2V power and an external 0.1uF capacitor.
10	RSATA	I	RSATA	It is required to connect 18KΩ resistor to ground for SATA-PHY.
11	PA15	I/O	PA15 /MISO0/SD13/RXD0_	PA15: General-purpose bidirectional digital I/O pin. MISO0: SPI0 serial data pin, host input/slave output. SD13: SD1 controller data line 3. RXD0_: RXD0 pin mapping of UART0.
12	PA14	I/O	PA14 /MOSI0/TXD0_/SD12	PA14: General-purpose bidirectional digital I/O pin. MOSI0: SPI0 serial data pin, host output/slave input. TXD0_: TXD pin mapping of UART0. SD12: SD1 controller data line 2.
13	PA13	I/O	PA13 /SCK0/SD11	PA13: General-purpose bidirectional digital I/O pin. SCK0: SPI0 serial clock pin, host output/slave input. SD11: SD1 controller data line 1.
14	PA12	I/O	PA12 /SD10/SCS/INT5	PA12: General-purpose bidirectional digital I/O pin. SD10: SD1 controller data line 0. SCS: Chip selection input pin of SPI0 slave. INT5: IO interrupt 5.
15	PA2	I/O	PA2 /MISO1/LED2/CMD3	PA2: General-purpose bidirectional digital I/O pin. MISO1: SPI1 serial data pin, host input; in SPI1 simplex mode, serial data input and output pin. LED2: LED serial data line 2. CMD3: SD3 controller command signal line.
16	PA1	I/O	PA1 /MOSI1/LED1/CMD2	PA1: General-purpose bidirectional digital I/O pin. MOSI1: SPI1 serial data pin, host output. LED1: LED serial data line 1. CMD2: SD2 controller command signal line.
17	PA0	I/O	PA0 /SCK1/LED0/CMD1	PA0: General-purpose bidirectional digital I/O pin. SCK1: SPI1 serial clock pin, host clock output. LED0: LED serial data line 0. CMD1: SD1 controller command signal line.
18	V33IO1	P	V33IO1	1 set of 3.3V power supplies for peripherals, which require an external 0.1uF capacitor.
19	PB7	I/O	PB7 /TXD0/SD33	PB7: General-purpose bidirectional digital I/O pin.

				TXD0: UART0 serial data output. SD33: SD3 controller data line 3.
20	PB6	I/O	PB6 /RTS/SD32	PB6: General-purpose bidirectional digital I/O pin. RTS: MODEM output signal of UART0; request to send. SD32: SD3 controller data line 2.
21	PB5	I/O	PB5 /DTR/SD31	PB5: General-purpose bidirectional digital I/O pin. DTR: MODEM output signal of UART0; data terminal is ready. SD31: SD3 controller data line 1.
22	PB4	I/O	PB4 /SD30/RXD0/INT6	PB4: General-purpose bidirectional digital I/O pin. SD30: SD3 controller data line 0. RXD0: UART0 serial data input. INT6: IO interrupt 6.
23	PB3	I/O	PB3 /SD23/DCD	PB3: General-purpose bidirectional digital I/O pin. SD23: SD2 controller data line 3. DCD: MODEM input signal of UART0; carrier detection.
24	PB2	I/O	PB2 /PWM5_/SD22/CAP1_ /RI	PB2: General-purpose bidirectional digital I/O pin. PWM5_: PWM5 function mapping. SD22: SD2 controller data line 2. CAP1_: CAP1 function mapping. RI: MODEM input signal of UART0; ring indicator.
25	PB1	I/O	PB1 /SD21/DSR	PB1: General-purpose bidirectional digital I/O pin. SD21: SD2 controller data line 1. DTR: MODEM input signal of UART0; data device is ready.
26	PB0	I/O	PB0 /SD20/CTS	PB1: General-purpose bidirectional digital I/O pin. SD20: SD2 controller data line 0. CTS: MODEM input signal of UART0; clear to send.
27	VCC33	P	VCC33	3.3V power of PLL module, requires an external 0.1uF capacitor.
28	XO	I/O	XO	Crystal oscillator inverted output.
29	XI	I	XI	Crystal oscillator input.
30	RUSB	I/O	RUSB	It is required to connect 12KΩ resistor to ground for USB-PHY.
31	UD+	USB	DP	USB bus D+data line.
32	UD-	USB	DN	USB bus D-data line.
33	V33IO3	P	V33IO3	3 sets of 3.3V power supplies for USB and peripherals, require an external 0.1uF capacitor.

34	GND	P	GND	Ground: common ground, 0V reference point.
35	VCC12	P	VCC12	Core power, must be connected to VCC12 and an external 0.1uF capacitor.
36	PB11	I/O	PB11 /PWM6_/CAP2_	PB11: General-purpose bidirectional digital I/O pin. PWM6_: PWM6 function mapping. CAP2_: CAP2 function mapping.
37	PB10	I/O	PB10 /SD3CK/DBGIO/INT7	PB10: General-purpose bidirectional digital I/O pin. SD3CK: SD3 clock output pin. DBGIO: Data input and output port of simulation and debugging interface. INT7: IO interrupt 7.
38	RST#	I	RST# /DBGCK	RST#: External reset input pin, active low, built-in pull-up resistor. DBGCK: Clock input port of simulation and debugging interface.
39	PA3	I/O	PA3 /SD2CK/PWM4/LED3 /CAP0/INT0	PA3: General-purpose bidirectional digital I/O pin. SD2CK: SD2 clock output pin. PWM4: Pulse width modulation output channel 4. LED3: LED serial data line 3. CAP0: Timer 0 capture input pin. INT0: IO interrupt 0.
40	PA4	I/O	PA4 /SD1CK/PWM0/LEDC /RXD3/INT1	PA4: General-purpose bidirectional digital I/O pin. SD1CK: SD1 clock output pin PWM0: Pulse width modulation output channel 0. LEDC: LED serial data line. RXD3: UART3 serial data input. INT1: IO interrupt 1.
41	PA5	I/O	PA5 /TXD3/PWM1	PA5: General-purpose bidirectional digital I/O pin. TXD3: UART3 serial data output. PWM1: Pulse width modulation output channel 1.
42	PA6	I/O	PA6 /SD0CK/RXD2/INT2	PA6: General-purpose bidirectional digital I/O pin. SD0CK: SD0 controller clock line output. RXD2: UART2 serial data input. INT2: IO interrupt 2.
43	PA7	I/O	PA7 /TXD2/CMD0	PA7: General-purpose bidirectional digital I/O pin. TXD2: UART2 serial data output. CMD0: SD0 controller command signal line.
44	PA8	I/O	PA8 /PWM2/SD00/RXD1	PA8: General-purpose bidirectional digital I/O pin. PWM2: Pulse width modulation output channel 2. SD00: SD0 controller data line 0. RXD1: UART1 serial data input.

45	PA9	I/O	PA9 TXD1/PWM3/SD01	PA9: General-purpose bidirectional digital I/O pin. TXD1: UART1 serial data output. PWM3: Pulse width modulation output channel 3. SDO1: SD0 controller data line 1.
46	PA10	I/O	PA10 /PWM5/SD02/CAP1 /INT3	PA10: General-purpose bidirectional digital I/O pin. PWM5: Pulse width modulation output channel 5. SDO2: SD0 controller data line 2. CAP1: Timer 1 capture input pin. INT3: IO interrupt 3.
47	PA11	I/O	PA11 /PWM6/SD03/CAP2 /INT4	PA11: General-purpose bidirectional digital I/O pin. PWM6: Pulse width modulation output channel 6. SDO3: SD0 controller data line 3. CAP2: Timer 2 capture input pin. INT4: IO interrupt 4.
48	V33IO2	P	V33IO2	2 sets of 3.3V power supplies for peripherals, which require an external 0.1uF capacitor.

ATTENTION:

(1). I: Input; O: Output; P: Power.

(2). SATA: SATA signal; USB: USB signal.

(3). The priority of multiplexing functions of the pins in the table are arranged in order of high to low (excluding the main function GPIO function)

(4). Pins 37, 39 and 40 of the chips whose lot number is less than 50591 do not support SDIO clock output, while output by pin 42.

Remarks:

In order to be compatible with the power system of external devices, CH568 will divide zones to manage the power of peripherals and IOs, and provide multiple sets of power pins. In the above pin description, the pins marked with different colors belong to different power domains, and the assignments are as follows:

Font color: Same as power VCC33

Font color: Same as power V33IO1

Font color: Same as power V33IO2

Font color: Same as power V33IO3

Chapter 2 System Structure and Memory

2.1 System Structure

The following figure shows the system structure block diagram of CH568 chip.

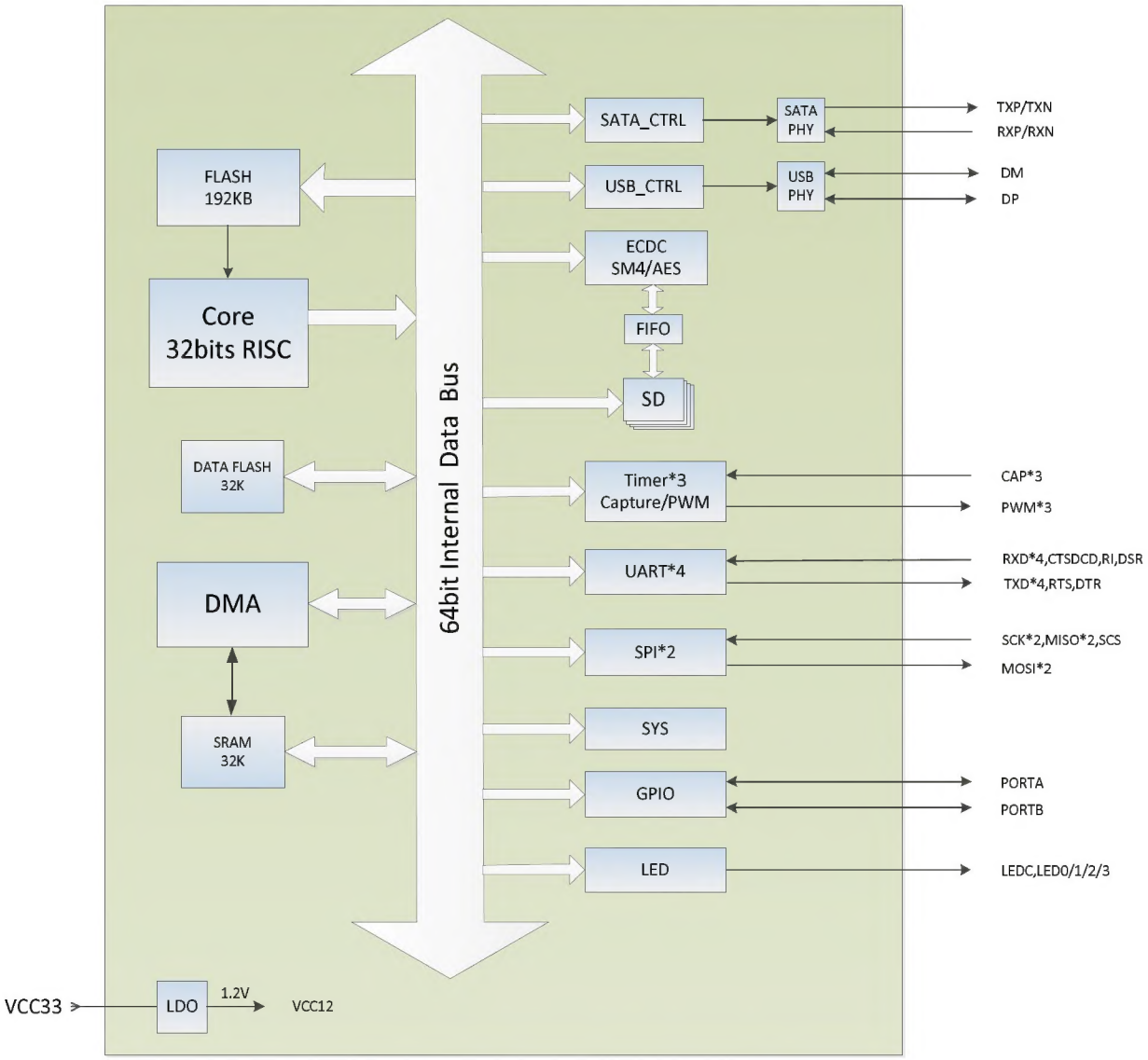


Figure 2-1 CH568 Internal Structure Block Diagram

The CPU core, DMA arbitration controller, SRAM and various peripheral modules are mounted on the 64bit system bus of CH568. The DMA controller can be used for peripheral modules such as USB, SATA, SD, SPI0, LED and TIMER.

2.2 Memory Mapping

CH568 contains a 4GB address space, and the memory map mainly contains several different areas, as shown in the figure below.

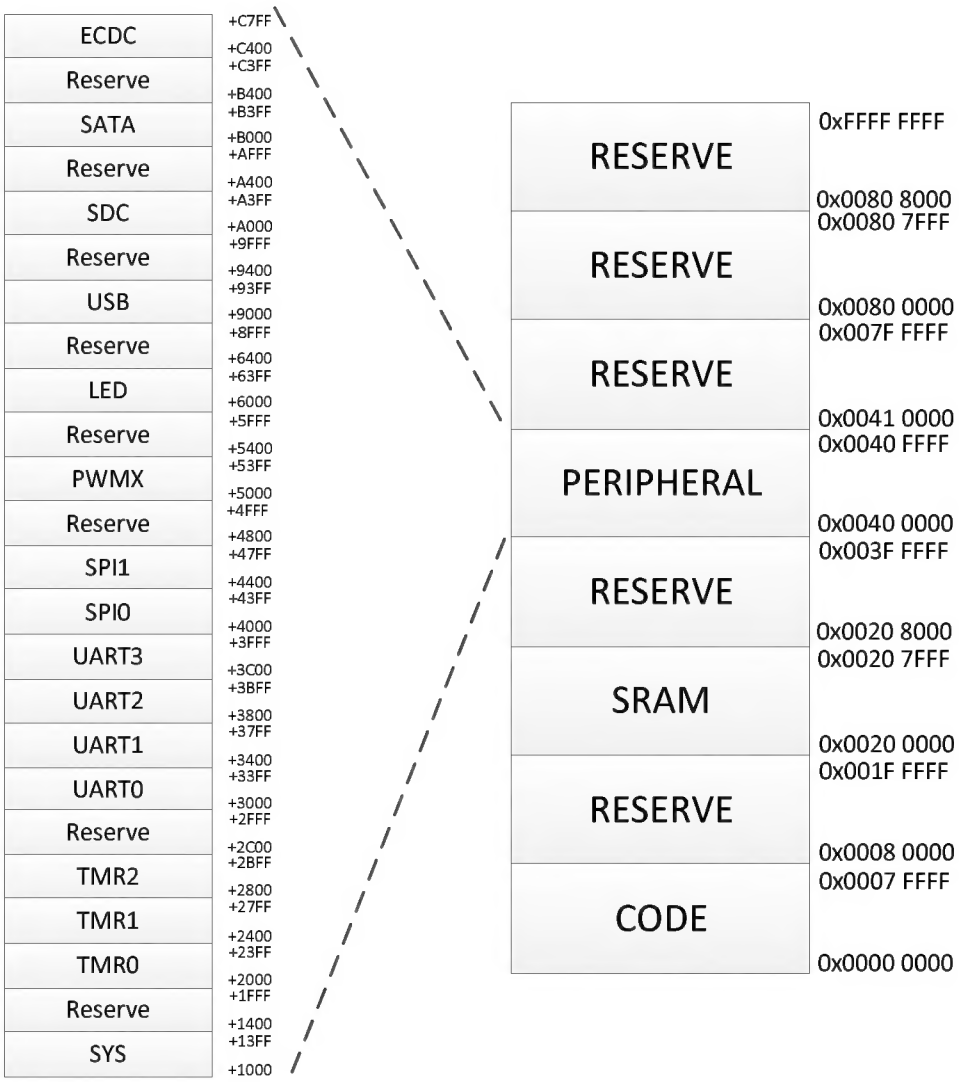


Figure 2-2 Memory Mapping

2.3 Memory Mapping

The address range of each memory mapping area is shown in the table below:

Table 2-1 Memory Mapping Area Address

Address range	Feature	Description
0x0000 0000-0x0007 FFFF	On-chip non-volatile memory	Flash memory (512KB)
0x0008 0000-0x001F FFFF	Reserved	-
0x0020 0000-0X0020 7FFF	On-chip SRAM, usually used to store data	32KB
0x0020 8000-0x003F FFFF	Reserved	-
0x0040 0000-0x0040 FFFF	Various peripherals	Multiple peripheral modules
0x0041 0000-0x007F FFFF	Reserved	-
0x0080 0000-0x0080 7FFF	Reserved	-
0x0080 8000-0xFFFF 7FFF	Reserved	-

2.4 Peripheral Address Assignment

CH568 mainly contains16 peripherals. Each peripheral occupies a certain address space, and the actual access address of peripheral register is: base address + offset address. In the following chapters, the address

of the register is described in detail. The following table shows the assignment of base address of each peripheral.

Table 2-2 Peripheral Base Address Assignment

Peripheral No.	Peripheral name	Peripheral base address
1	SYS	0x0040 1000
2	TMR0	0x0040 2000
3	TMR1	0x0040 2400
4	TMR2	0x0040 2800
5	UART0	0x0040 3000
6	UART1	0x0040 3400
7	UART2	0x0040 3800
8	UART3	0x0040 3C00
9	SPI0	0x0040 4000
10	SPI1	0x0040 4400
11	PWMX	0x0040 5000
12	LED	0x0040 6000
13	USB	0x0040 9000
14	SDC	0x0040 A000
15	SATA	0x0040 B000
16	ECDC	0x0040 C400

The following table shows the explanation of "Access" in the register description in the subsequent chapters:

Abbreviation	Description
RF	The read value is fixed, which is not affected by reset.
RO	Read only.
WO	Write only (the read value is 0 or invalid).
RZ	Read-only, automatically cleared after read operation.
WZ	Write to clear.
RW	Readable, writable.
RW1	Clear by reading/writing 1.
WA	Write-only (in safety mode), the read value is 0 or invalid.
RWA	Write in read/safety mode.

Chapter 3 System Control

3.1 Power Control

CH568 needs an external supply voltage of 3.3V. In the package pin description of Figure 1-1, multiple sets of power and ground are provided externally. Internal power management is provided with multi-power domain group mode, which can connect different power systems according to the peripheral resources used.

After the system or power is reset, CH568 is in running state. When the CPU does not need to continue to run, or some functional modules do not need to be used, the clock or independent power of these modules can be turned off, to reduce power consumption.

3.2 Reset Control

CH568 supports 3 types of reset forms, namely power-on reset, external manual reset and internal software reset. System will reload the configuration information and reload the program code into the RAM buffer after reset. The load time is about 8.8mS.

The register R8_GLOB_RESET_KEEP is only reset at the time of power-on reset, and it is not affected by other types of reset.

3.2.1 Power-on Reset

When the power voltage is lower than the power-on reset threshold Vpot, CH568 will be reset. The figure below shows the power-on reset of CH568.

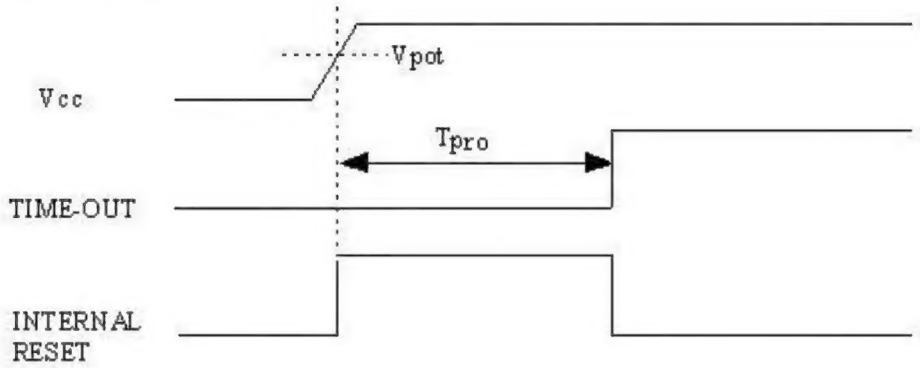


Figure 3-1 Power-on Reset

3.2.2 External Manual Reset

The external manual reset is generated by the low level applied to the RST# pin from the external. When the reset low level duration is greater than the minimum reset pulse width (Trst), the CH568 chip will be triggered to reset.

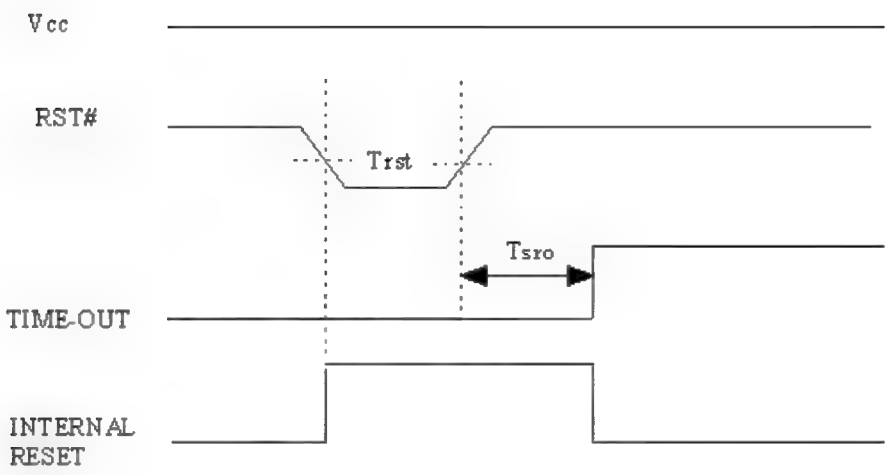


Figure 3-2 External Reset

3.2.3 Internal Software Reset

CH568 provides the internal software reset function, so that no external intervention is required to perform software reset in some specific situations. Set the bit RB_SOFTWARE_RESET of global reset configuration register (R8_RST_WDOG_CTRL) to 1, to realize software reset. This bit will be cleared automatically.

3.2.4 Reset Feature

Please refer to the timing parameter table in section 15.4 for reset property parameters.

3.3 Register Description

System control related register physical base address: 0x0040 1000

Table 3-1 List of Clock and CPU Control Related Registers

Name	Offset address	Description	Reset value
R8_SAFE_ACCESS_SIG	0x00	Safe access flag register	8h00
R8_CHIP_ID	0x01	Chip ID register	8h68
R8_SAFE_ACCESS_ID	0x02	Safe access ID register	8h02
R8_GLOB_ROM_CFG	0x04	ROM configuration register	8h80
R8_RST_BOOT_STAT	0x05	BOOT status register	8hC1
R8_RST_WDOG_CTRL	0x06	Reset register	8h00
R8_GLOB_RESET_KEEP	0x07	Reset keeping register	8h00
R8_SLP_WAKE_CTRL	0x0E	Wake-up control register	8h00
R8_SLP_POWER_CTRL	0x0F	Low-power management register	8h00

Safe access flag register (R8_SAFE_ACCESS_SIG)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SAFE_ACCESS_SIG	WO	Security access flag register. Some registers (access attribute is RWA) are protected and read/write operation can be conducted only after entering the safe access mode. Write 0x57 first and then write 0xA8 into this register, to enter the safe access mode, the time is limited to about 110 main clock cycles (Tsys), and it will be automatically protected beyond the limit.	00h
[6:4]	RB_SAFE_ACC_TIMER	RO	Safe access time, fixed at 128 Tsys	0

[1:0]	RB_SAFE_ACC_MODE	RO	Current safe access mode status: 11: Safe mode, register with RWA attribute can be accessed; Others: Non-safe mode;	0

Chip ID register (R8_CHIP_ID)

Bit	Name	Access	Description	Reset value
[7:0]	R8_CHIP_ID	RF	Fixed value of 68h, used to identify the chip.	68h

Safe access ID register (R8_SAFE_ACCESS_ID)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SAFE_ACCESS_ID	RF	Fixed value of 02h.	02h

ROM configuration register (R8_GLOB_ROM_CFG)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RWA	Reserved. For [7:6], must write 10b, and read value is 0.	0000b
3	RB_ROM_CODE_WE	RWA	Flash ROM code and data area erase/write enable bit: 1: Programmable/erasable; 0: Write protection	0
2	RB_ROM_DATA_WE	RWA	Flash ROM data area erase/write enable bit: 1: Programmable/erasable; 0: Write protection.	0
1	RB_CODE_RAM_WE	RWA	Code RAM area write enable bit: 1: Write enabled; 0: Write protection.	0
0	RB_ROM_EXT_RE	RO	External programmer read Flash ROM enable bit: 1: Read enabled; 0: Read protection.	0

BOOT status register (R8_RST_BOOT_STAT)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved.	11b
5	RB_BOOT_LOADER	RO	Boot loader status: 1: Boot loader status (Boot-Loader); 0: User program status.	0
4	RB_CFG_DEBUG_EN	RO	Debug enable control bit: 1: Enable; 0: Disable.	0
3	RB_CFG_BOOT_EN	RO	Boot loader enable control bit: 1: Enable; 0: Disable.	0
2	RB_CFG_RESET_EN	RO	External reset enable control bit: 1: Reset when the external inputs low-level	0

			signal; 0: Disable.	
[1:0]	RB_RESET_FLAG	RO	Last reset flag, as shown in Table 3-2.	1

Table 3-2 Last Reset Flag Description

RB_RESET_FLAG	Reset Flag Description
00b	Software reset, source: RB_SOFTWARE_RESET=1 and RB_BOOT_LOADER=0.
01b	Power on reset, source: Chip operating voltage is lower than the threshold voltage.
11b	Manual reset, source: RST# pin inputs low level.

Reset register (R8_RST_WDOG_CTRL)

Bit	Name	Access	Description	Reset value
[7:1]	Reserved	RO	Reserved. For [7:6], must write 01b.	00h
0	RB_SOFTWARE_RESET	WA/ WZ	System software reset, cleared automatically: 1: System reset; 0: No action	0

Reset hold register (R8_GLOB_RESET_KEEP)

Bit	Name	Access	Description	Reset value
[7:0]	R8_GLOB_RESET_KEEP	RW	Reset hold register. The value of this register is not affected by manual reset, software reset or watchdog reset.	00h

Wake-up control register (R8_SLP_WAKE_CTRL)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	0
4	RB_SLP_GPIO_WAKE	RWA	GPIO port wake-up enable control bit: 1: Enable; 0: Disable.	0
3	Reserved	RO	Reserved.	0
2	RB_SLP_SATA_WAKE	RWA	SATA wake-up enable control bit: 1: Enable; 0: Disable.	0
1	RB_SLP_USB1_WAKE	RWA	USB1 wake-up enable control bit: 1: Enable; 0: Disable.	0
0	Reserved	RO	Reserved.	0

Low-power management register (R8_SLP_POWER_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_SLP_STANDBY	RWA	Low-power mode control bit, cleard automatically after entering: 1: Request core low power; 0: No action.	0
[6:3]	Reserved	RO	Reserved.	0
2	RB_SLP_SATA_PWRDN	RWA	SATA power control bit: 1: Power off; 0: Power on normally.	0

1	RB_SLP_USB1_PWRDN	RWA	USB power control bit: 1: Power off; 0: Power on normally.	0
0	Reserved	RO	Reserved.	0

3.4 Low Power Mode and Wake-up

In the low-power state (RB_SLP_STANDBY bit is set to 1), the PLL will stop working, the internal clock of CH568 will be suspended, the CPU will not work or respond to any interrupts. However, CPU starts working after waking up, and it finds that the wake-up event is also an interrupt event (for example, a certain GPIO wakes up and GPIO interrupt is generated), it will treate it as an interrupt.

In order to reduce power consumption, the physical PHY module (such as USB/SATA) that is not used during the low power period shall be turned off before entering the low power state. Set the RB_SLP_SATA_PWRDN bit to 1, and the RB_SLP_USB1_PWRDN bit to 1. In addition, GPIO pins cannot be in a floating state, and need to be set to output state or external input state with a fixed level. If there is no external input with a fixed level, it needs to be set to the input state in internal pull-down mode.

In low power mode, CH568 only supports part of GPIOs or USB or SATA wake up, please refer to R8_SLP_WAKE_CTRL register.

There are 8 GPIO pins that support wake-up, which are 8 pins that support GPIO interrupts. GPIO wake-up event source is the same as the GPIO interrupt event source. However, only when there is level trigger (R8_GPIO_INT_MODE is not required) and the bit corresponding to R8_GPIO_INT_POLAR is 0, the GPIO pin will wake up in a low level. When the bit corresponding to R8_GPIO_INT_POLAR is 1, GPIO will wake up in a high level.

Take GPIO port PA3 wake-up as an example, the configuration is as follows:

```
RB_GPIO_PA3_IP=0;
RB_GPIO_PA3_IE=1;
RB_SLP_GPIO_WAKE=1;
```

When there is a low level at PA3 port, a wake-up event will be generated. After CH568 exits the low-power mode, it will trigger the GPIO interrupt of PA3 port.

The followings must be configured when USB wakes up:

```
RB_SLP_USB1_WAKE = 1;
RB_SLP_USB1_PWRDN = 1;      //In safe mode
bUH_TX_BUS_SUSPEND = 1;
bUH_TX_BUS_SUSPEND = 0;
```

When the wake-up signal appears on the USB port, a wake-up event will be generated. After CH568 exits the low-power mode, the following register needs to be set.

```
RB_SLP_USB1_PWRDN = 0;      //In safe mode
bUH_TX_BUS_RESUME = 1;
bUH_TX_BUS_RESUME = 0;
```

Chapter 4 Clock Control

4.1 Clock Block Diagram

The internal clock structure of CH568 is shown in the figure below:

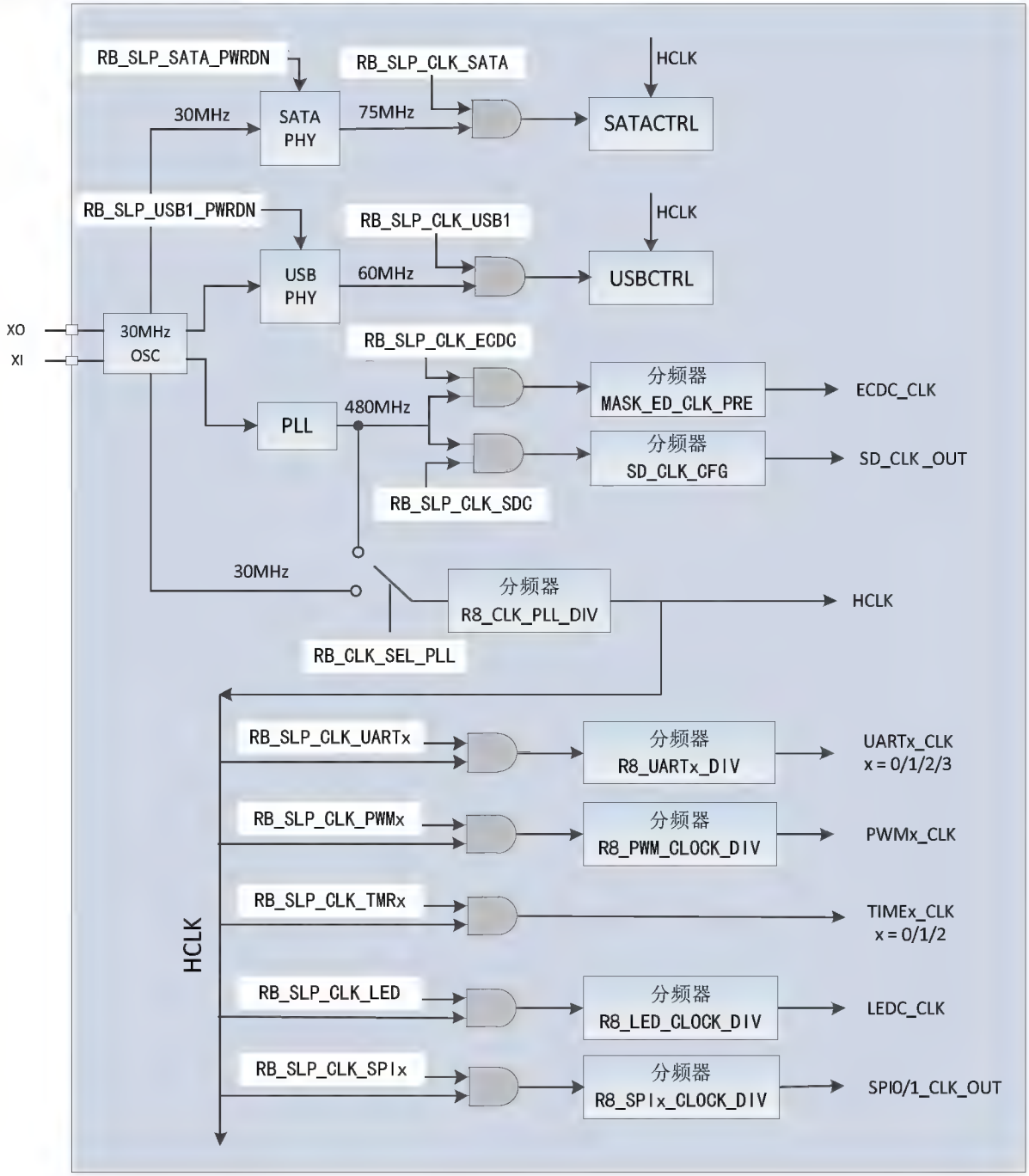


Figure 4-1 Clock Structure Block Diagram

After the external clock is sent to the CH568, it will be connected to USB-PHY and SATA-PHY to generate the clock frequency required by USB and SATA controllers, and generate the frequency multiplication clock of 480MHz through the PLL module. Get the clock frequency of 30MHz or 480MHz before frequency division through the clock source selection control bit (RB_CLK_SEL_PLL). The system clock Fsys (HCLK) can be obtained after the clock frequency is divided by R8_CLK_PLL_DIV, namely the main clock of CPU,

with a range of 2MHz- 120MHz.

Each peripheral module clock has a corresponding clock register control bit, which can be turned on or off individually. In order to reduce the power consumption of chip, you can turn off the function module clocks that are not used.

4.2 Register Description

Clock control related register physical base address: 0x0040 1000

Table 4-1 List of Clock Control Related Registers

Name	Offset address	Description	Reset value
R8_CLK_PLL_DIV	0x08	PLL output clock divider register	8h42
R8_CLK_CFG_CTRL	0x0A	Clock configuration register	8h80
R8_SLP_CLK_OFF0	0x0C	Sleep control register 0	8h00
R8_SLP_CLK_OFF1	0x0D	Sleep control register 1	8h00

PLL output clock divider register (R8_CLK_PLL_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_CLK_PLL_DIV	RWA	The lower 4 bits are valid. For [7:6], must write 01b, and the minimum value is 2.	8h42

Clock configuration register (R8_CLK_CFG_CTRL)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RWA	Reserved. Must write 10b to [7:6].	100000b
1	RB_CLK_SEL_PLL	RWA	Clock source selection: 1:PLL 480MHz; 0: External crystal oscillator of 30MHz.	0
0	RB_CLK_PLL_SLEEP	RWA	PLL sleep control bit: 1: PLL sleep; 0: PLL is working normally.	0

Sleep control register 0 (R8_SLP_CLK_OFF0)

Bit	Name	Access	Description	Reset value
7	RB_SLP_CLK_UART3	RWA	UART3 clock control bit: 1: UART3 clock is turned off; 0: UART3 clock is turned on.	0
6	RB_SLP_CLK_UART2	RWA	UART2 clock control bit: 1: UART2 clock is turned off; 0: UART2 clock is turned on.	0
5	RB_SLP_CLK_UART1	RWA	UART1 clock control bit: 1: UART1 clock is turned off; 0: UART1 clock is turned on.	0
4	RB_SLP_CLK_UART0	RWA	UART0 clock control bit: 1: UART0 clock is turned off; 0: UART0 clock is turned on.	0
3	RB_SLP_CLK_PWMX	RWA	PWM clock control bit: 1: PWM clock is turned off;	0

			1: PWM clock is turned on.	
2	RB_SLP_CLK_TMR2	RWA	TIMER2 clock control bit: 1: TIMER2 clock is turned off; 0: TIMER2 clock is turned on.	0
1	RB_SLP_CLK_TMR1	RWA	TIMER1 clock control bit: 1: TIMER1 clock is turned off; 0: TIMER1 clock is turned on.	0
0	RB_SLP_CLK_TMR0	RWA	TIMER0 clock control bit: 1: TIMER0 clock is turned off; 0: TIMER0 clock is turned on.	0

Sleep control register 1 (R8_SLP_CLK_OFF1)

Bit	Name	Access	Description	Reset value
7	RB_SLP_CLK_ECDC	RWA	ECDC (encryption and decryption module) clock control bit: 1: ECDC clock is turned off; 1: ECDC clock is turned on.	0
6	RB_SLP_CLK_SATA	RWA	SATA clock control bit: 1: SATA clock is turned off; 0: SATA clock is turned on.	0
5	RB_SLP_CLK_USB1	RWA	USB clock control bit: 1: USB clock is turned off; 1: USB clock is turned on.	0
4	Reserved	RO	Reserved.	0
3	RB_SLP_CLK_LED	RWA	LEDC clock control bit: 1: LEDC clock is turned off; 1: LEDC clock is turned on.	0
2	RB_SLP_CLK_SDC	RWA	SDC clock control bit: 1: SDC clock is turned off; 1: SDC clock is turned on.	0
1	RB_SLP_CLK_SPI1	RWA	SPI1 clock control bit: 1: SPI1 clock is turned off; 0: SPI1 clock is turned on.	0
0	RB_SLP_CLK_SPI0	RWA	SPI0 clock control bit: 1: SPI0 clock is turned off; 0: SPI0 clock is turned on.	0

4.3 System Clock Configuration

External crystal oscillator clock: Fosc = 30MHz;
PLL frequency multiplication clock: Fpll = 480MHz;

1. Select the PLL clock source: Fsrc = RB_CLK_SEL_PLL? PLL_FREQ : OSC_FREQ;
2. System clock calculation: Fsys = SRC_FREQ / R8_CLK_PLL_DIV, (2MHz-120MHz).

When the system is powered on, 30MHz is selected as the PLL clock source by default, the frequency division factor is 2, and the default main frequency is 15MHz.

Chapter 5 General and Multiplexing Functions I/O

5.1 Introduction to GPIO

The system is equipped with 2 sets of GPIOs (PA and PB), with a total of 26 general-purpose input and output pins, and some pins have multiplexing and mapping functions. Each GPIO port has a 32-bit direction configuration register (R32_Px_DIR), a 32-bit data input register (R32_Px_PIN), a 32-bit data output register (R32_Px_OUT), a 32-bit data clear register (R32_Px_CLR), a 32-bit pull-up configuration register (R32_Px_PU), a 32-bit open drain output and input pull-down configuration register (R32_Px_PD), a 32-bit I/O drive capability configuration register (R32_Px_DRV) and a 32-bit Schmitt trigger enable configuration register (R32_Px_SMT).

In the PA port, bits PA[0]-PA[15] are valid, corresponding to the 16 GPIO pins on the chip. In the PB port, the PB[0]-PB[7] and PB[10]-PB[11] bits are valid, corresponding to the 10 GPIO pins on the chip. Among them, 8 GPIO ports have interrupt function and can realize sleep wake-up function.

Each I/O port bit can be freely programmed, but the I/O port register must be accessed by 8-bit, 16-bit or 32-bit words. If the multiplexing function of pin is not enabled, it will be used as a general-purpose I/O port.

The following figure is a block diagram of the internal structure of GPIO:

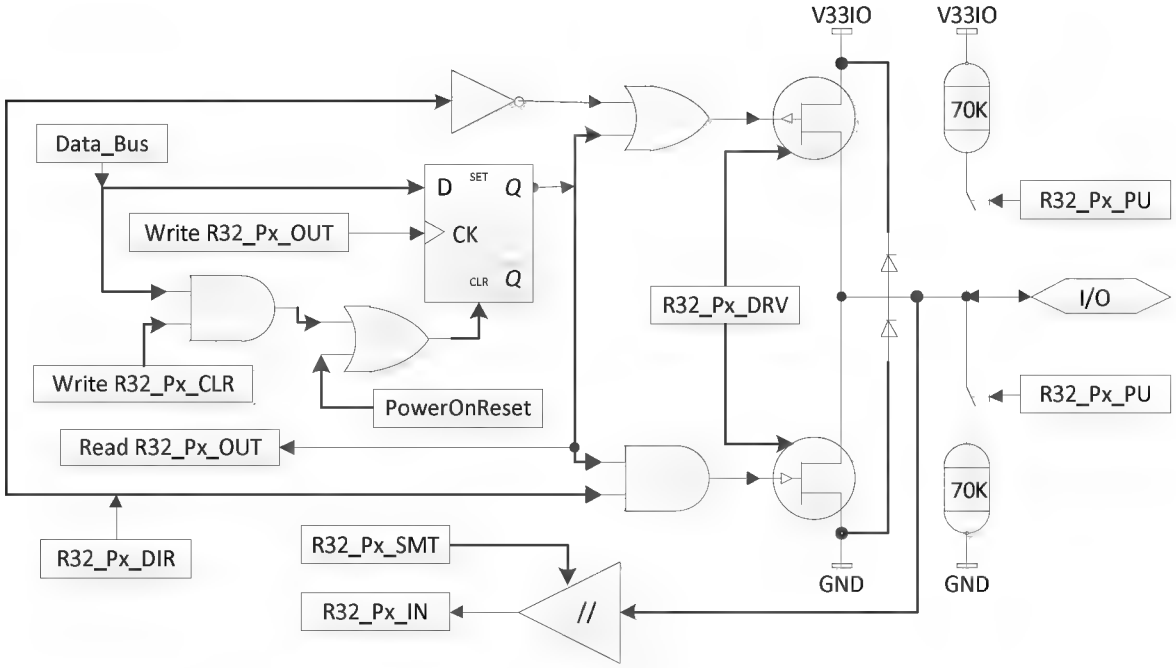


Figure 5-1 IO Internal Structure Block Diagram

5.2 Register Description

GPIO related register physical base address: 0x0040 1000

Table 5-1 List of GPIO Related Registers

Name	Offset address	Description	Reset value
R8_GPIO_INT_STATUS	0x1C	GPIO interrupt flag register	8h00
R8_GPIO_INT_ENABLE	0x1D	GPIO interrupt enable register	8h00
R8_GPIO_INT_MODE	0x1E	GPIO interrupt trigger mode register	8h00
R8_GPIO_INT_POLAR	0x1F	GPIO interrupt polarity register	8h00

R32_PA_DIR	0x40	PA port direction setting register	32h0000 0000
R32_PA_PIN	0x44	PA port data input register	32hxxxx xxxx
R32_PA_OUT	0x48	PA port data output register	32h0000 0000
R32_PA_CLR	0x4C	PA port output clear register	32h0000 0000
R32_PA_PU	0x50	PA port pull-up enable register	32h0000 0000
R32_PA_PD	0x54	PA port open-drain output and input pull-down configuration register	32h0000 0000
R32_PA_DRV	0x58	PA port drive capability configuration register	32h0000 0000
R32_PA_SMT	0x5C	PA port Schmitt trigger enable configuration register	32h0000 0000
R32_PB_DIR	0x60	PB port direction setting register	32h0000 0000
R32_PB_PIN	0x64	PB port data input register	32hxxxx 8000
R32_PB_OUT	0x68	PB port data output register	32h0000 0000
R32_PB_CLR	0x6C	PB port output clear register	32h0000 0000
R32_PB_PU	0x70	PB port pull-up configuration register	32h0000 0000
R32_PB_PD	0x74	PB port open-drain output and input pull-down configuration register	32h0000 0000
R32_PB_DRV	0x78	PB port drive capability configuration register	32h0000 0000
R32_PB_SMT	0x7C	PB port Schmitt trigger enable configuration register	32h0000 0000
R8_PORT_PIN	0x12	Multiplexing and remapping configuration register	8h00

GPIO interrupt flag register (R8_GPIO_INT_STATUS)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_PB10_IS	RW1	PB10 pin interrupt flag bit, write 1 to clear: 1: An interrupt is generated; 0: No interrupt is generated.	0
6	RB_GPIO_PB4_IS	RW1	PB4 pin interrupt flag bit, write 1 to clear: 1: An interrupt is generated; 0: No interrupt is generated.	0
5	RB_GPIO_PA12_IS	RW1	PA12 pin interrupt flag bit, write 1 to clear: 1: An interrupt is generated; 0: No interrupt is generated.	0
4	RB_GPIO_PA11_IS	RW1	PA11 pin interrupt flag bit, write 1 to clear: 1: An interrupt is generated; 0: No interrupt is generated.	0
3	RB_GPIO_PA10_IS	RW1	PA10 pin interrupt flag bit, write 1 to clear: 1: An interrupt is generated; 0: No interrupt is generated.	0
2	RB_GPIO_PA6_IS	RW1	PA6 pin interrupt flag bit, write 1 to clear: 1: An interrupt is generated; 0: No interrupt is generated.	0
1	RB_GPIO_PA4_IS	RW1	PA4 pin interrupt flag bit, write 1 to clear: 1: An interrupt is generated; 0: No interrupt is generated.	0

0	RB_GPIO_PA3_IS	RW1	PA3 pin interrupt flag bit, write 1 to clear: 1: An interrupt is generated; 0: No interrupt is generated.	0

GPIO interrupt enable register (R8_GPIO_INT_ENABLE)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_PB10_IE	RW	PB10 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
6	RB_GPIO_PB4_IE	RW	PB4 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
5	RB_GPIO_PA12_IE	RW	PA12 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
4	RB_GPIO_PA11_IE	RW	PA11 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	RB_GPIO_PA10_IE	RW	PA10 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	RB_GPIO_PA6_IE	RW	PA6 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	RB_GPIO_PA4_IE	RW	PA4 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	RB_GPIO_PA3_IE	RW	PA3 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

GPIO interrupt trigger mode register (R8_GPIO_INT_MODE)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_PB10_IM	RW	PB10 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
6	RB_GPIO_PB4_IM	RW	PB4 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
5	RB_GPIO_PA12_IM	RW	PB12 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
4	RB_GPIO_PA11_IM	RW	PA11 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
3	RB_GPIO_PA10_IM	RW	PA10 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0

2	RB_GPIO_PA6_IM	RW	PA6 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
1	RB_GPIO_PA4_IM	RW	PA4 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
0	RB_GPIO_PA3_IM	RW	PA3 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0

GPIO interrupt polarity register (R8_GPIO_INT_POLAR)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_PB10_IP	RW	PB10 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
6	RB_GPIO_PB4_IP	RW	PB4 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
5	RB_GPIO_PA12_IP	RW	PB12 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
4	RB_GPIO_PA11_IP	RW	PA11 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
3	RB_GPIO_PA10_IP	RW	PA10 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
2	RB_GPIO_PA6_IP	RW	PA6 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
1	RB_GPIO_PA4_IP	RW	PA4 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
0	RB_GPIO_PA3_IP	RW	PA3 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0

PA port direction setting register (R32_PA_DIR)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_DIR	RW	Current input and output direction control of PA pin: 1: The pin direction is output mode; 0: The pin direction is input mode.	0

PA port input data register (R32_PA_PIN)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0

[15:0]	R32_PA_PIN	RO	PA pin level status: 1: Pin inputs high level; 0: Pin inputs low level; The value of this bit is valid only when the corresponding bit of the direction register (R32_PA_DIR) is 0.	0

PA port output data register (R32_PA_OUT)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_OUT	RW	PA pin output level status: 1: The pin outputs high level; 0: The pin outputs low level; The value of this bit is valid only when the corresponding bit of the direction register (R32_PA_DIR) is 1.	0

PA port bit clear register (R32_PA_CLR)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_CLR	WZ	PA hold/clear data output control: 1: The pin outputs low level; 0: No effect.	0

PA port pull-up configuration register (R32_PA_PU)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_PU	RW	PA pin pull-up function enable control: 1: Enable pin pull-up function; 0: Disable pin pull-up function.	0

PA port open-drain output and input pull-down configuration register (R32_PA_PD)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_PD	RW	When the corresponding bit of direction register (R32_PA_DIR) is configured as 1 (ie output mode): 1: Enable open-drain output function of this pin; 0: Disable open-drain output function of this pin; When the corresponding bit of direction register (R32_PA_DIR) is configured as 0 (ie input mode): 1: Enable the pull-down function of this pin; 0: Disable the pull-down function of this pin.	0

PA port drive capability configuration register (R32_PA_DRV)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_DRV	RW	PA pin output drive capability control: 1: The maximum driving current is 16mA; 0: The maximum drive current is 8mA.	0

PA port Schmitt trigger enable configuration register (R32_PA_SMT)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_SMT	RW	PA pin Schmitt trigger function control: 1: Enable the Schmitt trigger input function or low slope output function of this pin; 0: Disable the Schmitt trigger input function or low slope output function of this pin.	1

PB port direction setting register (R32_PB_DIR)

Bit	Name	Access	Description	Reset value
[31:12] [9:8]	Reserved	RO	Reserved.	0
[11:10] [7:0]	R32_PB_DIR	RW	Current input and output direction control of PB pin: 1: The pin direction is output mode; 0: The pin direction is input mode.	0

PB port input data register (R32_PB_PIN)

Bit	Name	Access	Description	Reset value
[31:12] [9:8]	Reserved	RO	Reserved.	0
[11:10] [7:0]	R32_PB_PIN	RO	Current level status of PB pin: 1: Pin inputs high level; 0: Pin inputs low level. The value of this bit is valid only when the corresponding bit of the direction register (R32_PB_DIR) is 0.	0

PB port output data register (R32_PB_OUT)

Bit	Name	Access	Description	Reset value
[31:12] [9:8]	Reserved	RO	Reserved.	0
[11:10] [7:0]	R32_PB_OUT	RW	PB pin output level status: 1: The pin outputs high level; 1: The pin outputs low level. The value of this bit is valid only when the corresponding bit of the direction register (R32_PB_DIR) is 1.	0

PB port bit clear register (R32_PB_CLR)

Bit	Name	Access	Description	Reset value
[31:12] [9:8]	Reserved	RO	Reserved.	0
[11:10] [7:0]	R32_PB_CLR	WZ	PB hold/clear data output control: 1: The pin outputs low level; 0: No effect.	0

PB port pull-up configuration register (R32_PB_PU)

Bit	Name	Access	Description	Reset value
[31:12] [9:8]	Reserved	RO	Reserved.	0
[11:10] [7:0]	R32_PB_PU	RW	PB pin pull-up function enable control: 1: Enable pin pull-up function; 0: Disable pin pull-up function.	0

PB port open-drain output and input pull-down configuration register (R32_PB_PD)

Bit	Name	Access	Description	Reset value
[31:12] [9:8]	Reserved	RO	Reserved.	0
[11:10] [7:0]	R32_PB_PD	RW	When the corresponding bit of direction register (R32_PB_DIR) is configured as 1 (ie output mode): 1: Enable the open-drain output function of this pin; 1: Disable the open-drain output function of this pin; When the corresponding bit of direction register (R32_PB_DIR) is configured as 0 (ie input mode): 1: Enable the pull-down function of this pin; 1: Disable the pull-down function of this pin; If the corresponding bit configuration of the pull-up configuration register (R32_PB_PU) is also 1, the input status weak holding function is enabled.	0

PB port drive capability configuration register (R32_PB_DRV)

Bit	Name	Access	Description	Reset value
[31:12] [9:8]	Reserved	RO	Reserved.	0
[11:10] [7:0]	R32_PB_DRV	RW	PB pin output drive capability control: 1: The maximum driving current is 16mA; 0: The maximum drive current is 8mA.	0

PB port Schmitt trigger enable configuration register (R32_PB_SMT)

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0

[9:8]				
[11:10] [7:0]	R32_PB_SMT	RW	PB pin Schmitt trigger function control: 1: Enable the Schmitt trigger input function or low slope output function of this pin; 0: Disable the Schmitt trigger input function or low slope output function of this pin.	1

Multiplexing and remapping configuration register (R8_PORT_PIN)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	0
4	RB_PIN_UART0	RW	UART0 remapping configuration bit: 1: RXD0/TXD0 to PA15/PA14 pin; 0: RXD0/TXD0 to PB4/PB7 pin.	0
3	Reserved	RO	Reserved.	0
2	RB_PIN_TMR2	RW	TIMER2 remapping configuration bit: 1: TMR2/PWM6/CAP2 to PB11 pin; 0: TMR2/PWM6/CAP2 to PA11 pin.	0
1	RB_PIN_TMR1	RW	TIMER1 remapping configuration bit: 1: TMR1/PWM5/CAP1 to PB2 pin; 0: TMR1/PWM5/CAP1 to PA10 pin.	0
0	Reserved	RO	Reserved.	0

5.3 GPIO Multiplexing and Remapping

5.3.1 Multiplexing Function

Some I/O pins of chip have the function of multiplexing. After power on, all I/O pins have common I/O functions by default. After enabling different functional modules, the corresponding pins are configured as corresponding functional pins of each functional module.

If a pin is multiplexed with multiple functions, and multiple functions are enabled, please refer to the function order in the "Multiplexing Function and Mapping" list in the pin of section 1.2 for the priority order of multiplexing function.

For example: If PA0 pin is multiplexed as /SCK1/LED0/CMD1, the clock function of SPI1 has priority, and the CMD1 function of SD1 controller is the lowest. In this way, the multiplexing functions with the relatively higher priority of the pin whose functions with the lowest priority need not to be used can be enabled among multiple multiplexing functions.

The following tables list the I/O pins used by each functional module.

Table 5-2 Serial Peripheral Interface (SPI0)

Pin	GPIO	Function Description
SCS	PA12	Chip selection input pin of SPI0 slave
SCK0	PA13	SPI0 serial clock pin, host output/slave input
MOSI0	PA14	SPI0 serial data pin, host output/slave input
MISO0	PA15	SPI0 serial data pin, host input/slave output

Table 5-3 Serial Peripheral Interface (SPI1)

Pin	GPIO	Function Description
SCK1	PA0	Output pin of SPI1 serial clock

MOSI1	PA1	SPI1 serial data output pin (only host function)
MISO1	PA2	SPI1 serial data input pin (only host function)

Table 5-4 Universal Asynchronous Receiver/Transmitter (UART0)

Pin	GPIO	Function Description
RXD0	PB4	UART0 receiver input pin
TXD0	PB7	UART0 transmitter output pin
RXD0_	PA15	RXD pin function mapping of UART0
TXD0_	PA14	TXD pin function mapping of UART0
DTR	PB5	MODEM signal of UART0, data terminal ready
RTS	PB6	MODEM signal of UART0, request to send
CTS	PB0	MODEM signal of UART0, clear to send
DSR	PB1	MODEM signal of UART0, data device is ready
RI	PB2	MODEM signal of UART0, ring indicator
DCD	PB3	MODEM signal of UART0, carrier detection

Table 5-5 Universal Asynchronous Receiver/Transmitter (UART1-3)

Pin	GPIO	Function Description
RXD1	PA8	UART1 receiver input pin
TXD1	PA9	UART1 transmitter output pin
RXD2	PA6	UART2 receiver input pin
TXD2	PA7	UART2 transmitter output pin
RXD3	PA4	UART3 receiver input pin
TXD3	PA5	UART3 transmitter output pin

Table 5-6 SD Controller (SD0-3)

Pin	GPIO	Function Description
SDCK	PA6	SD0-3 clock signal pin
CMD0	PA7	SD0 command signal pin
SD00	PA8	SD0 data signal 0 pin
SD01	PA9	SD0 data signal 1 pin
SD02	PA10	SD0 data signal 2 pin
SD03	PA11	SD0 data signal 3 pin
CMD1	PA0	SD1 data signal pin
SD10	PA12	SD1 data signal 0 pin
SD11	PA13	SD1 data signal 1 pin
SD12	PA14	SD1 data signal 2 pin
SD13	PA15	SD1 data signal 3 pin
CMD2	PA1	SD2 command signal pin
SD20	PB0	SD2 command signal 0 pin
SD21	PB1	SD2 command signal 1 pin
SD22	PB2	SD2 command signal 2 pin
SD23	PB3	SD2 command signal 3 pin
CMD3	PA2	SD3 command signal pin
SD30	PB4	SD3 command signal 0 pin
SD31	PB5	SD3 command signal 1 pin
SD32	PB6	SD3 command signal 2 pin

SD33	PB7	SD3 command signal 3 pin

Table 5-7 LED Control Card

Pin	GPIO	Function Description
LED0	PA0	LED serial data 0 pin
LED1	PA1	LED serial data 1 pin
LED2	PA2	LED serial data 2 pin
LED3	PA3	LED serial data 3 pin
LEDC	PA4	LED serial clock pin

Table 5-8 ISP Download (ISP)

Pin	GPIO	Function Description
SCS	PA12	ISP download chip select input pin
SCK	PA13	ISP download clock input pin
MOSI	PA14	ISP download data input pin
MISO	PA15	ISP download data output pin
RST	RST#	ISP download reset input pin

5.3.2 Remapping

In order to optimize the number of peripherals in the chip package, some multiplexing functions can be remapped to other pins. The remapping of pin can be realized by means of setting the multiplexing and mapping register R8_PORT_PIN.

CH568 supports the remapping of UART0, TIMER1 and TIMER 2 peripheral pins, please refer to the following table for details:

Table 5-9 Remapping Pin

Peripheral Function	Default Pin	Remapping Pin
UART0	PB4/PB7	PA15/PA14
TIMER1/PWM5	PA10	PB2
TIMER2/PWM6	PA11	PB11

Chapter 6 Serial Peripheral Interface (SPI)

6.1 Introduction to SPI

SPI is a full-duplex serial interface. It can handle multiple masters and slaves connected to the specified bus. During data communication, there can only be one master and one slave for communication on the bus. Usually SPI interface consists of 4 pins: SPI chip selection pin (SCS), SPI clock pin (SCK), SPI serial data pin MISO (master input/slave output pin) and SPI serial data pin MOSI (master output/slave input pin).

The CH568 chip has 2 SPI interfaces, and their respective characteristics are as follows:
SPI0 features:
(1). Support master mode and device mode;
(2). Compatible with Serial Peripheral Interface (SPI) specification;
(3). Support the data transmission methods in mode 0 and mode 3;
(4). 8-bit data transmission mode;
(5). The clock frequency is close to half of Fsys;
(6). 8-byte FIFO;
(7). The device mode supports the first byte as command mode or data stream mode;
(8). Support DMA data transmission.

SPI1 features:
(1). Only support master mode;
(2). Support the data transmission methods in mode 0 and mode 3;
(3). 8-bit data transmission mode;
(4). The maximum clock frequency is close to half of Fsys;
(5). 8-byte FIFO.

6.2 Register Description

SPI0 related register physical start address: 0x0040 4000
SPI1 related register physical start address: 0X0040 4400

Table 6-1 List of SPI0 Related Registers

Name	Offset address	Description	Reset value
R8_SPI0_CTRL_MOD	0x00	SPI0 mode configuration register	8h00
R8_SPI0_CTRL_CFG	0x01	SPI0 configuration register	8h00
R8_SPI0_INTER_EN	0x02	SPI0 interrupt enable register	8h00
R8_SPI0_CLOCK_DIV R8_SPI0_SLAVE_PRE	0x03	SPI0 clock frequency division register in master mode SPI0 preset data register in device mode	8h10
R8_SPI0_BUFFER	0x04	SPI0 data buffer	8hxx
R8_SPI0_RUN_FLAG	0x05	SPI0 working status register	8h00
R8_SPI0_INT_FLAG	0x06	SPI0 interrupt flag register	8h00
R8_SPI0_FIFO_COUNT	0x07	SPI0 transceiver FIFO counter register	8hxx
R16_SPI0_TOTAL_CNT	0x0C	SPI0 transceiver data length register	16hxxxx
R8_SPI0_FIFO	0x10	SPI0 FIFO register	8hxx
R8_SPI0_FIFO_COUNT1	0x13	SPI0 transceiver FIFO counter register	8hxx
R16_SPI0_DMA_NOW	0x14	Current address of SPI0 DMA buffer	16hxxxx

R16_SPI0_DMA_BEG	0x18	Start address of SPI0 DMA buffer	16hxxxx
R16_SPI0_DMA_END	0x1C	End address of SPI0 DMA buffer	16hxxxx

Table 6-2 List of SPI1 Related Registers

Name	Offset address	Description	Reset value
R8_SPI1_CTRL_MOD	0x00	SPI1 mode configuration register	8h00
R8_SPI1_CTRL_CFG	0x01	SPI1 configuration register	8h00
R8_SPI1_INTER_EN	0x02	SPI1 interrupt enable register	8h00
R8_SPI1_CLOCK_DIV	0x03	SPI1 clock frequency division register in master mode	8hxx
R8_SPI1_BUFFER	0x04	SPI1 data buffer	8hxx
R8_SPI1_RUN_FLAG	0x05	SPI1 working status register	8h00
R8_SPI1_INT_FLAG	0x06	SPI1 interrupt flag register	8h00
R8_SPI1_FIFO_COUNT	0x07	SPI1 transceiver FIFO counter register	8hxx
R16_SPI1_TOTAL_CNT	0x0C	SPI1 transceiver data length register	16hxxxx
R8_SPI1_FIFO	0x10	SPI1 FIFO register	8hxx
R8_SPI1_FIFO_COUNT1	0x13	SPI1 transceiver FIFO counter register	8hxx

SPI mode configuration register (R8_SPIx_CTRL_MOD) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_SPI_MISO_OE	RW	MISO pin output enable bit (can be used in data line switching direction of 2-wire mode): 1: MISO pin output enabled; 0: MISO pin output disabled.	0
6	RB_SPI_MOSI_OE	RW	MOSI pin output enable bit: 1: MISO pin output enabled; 0: MISO pin output disabled.	0
5	RB_SPI_SCK_OE	RW	SCK pin output enable bit 1: SCK pin output enabled; 0: SCK pin output disabled.	0
4	RB_SPI_FIFO_DIR	RW	FIFO direction setting bit: 1: Input mode (read data in host mode); 0: Output mode (write data in host mode).	0
3	RB_SPI_SLV_CMD_MOD	RW	SPI0 device mode first byte configuration bit, only for SPI0: 1: First byte command mode; 0: Data flow mode. In the first byte command mode, it will be regarded as a command code when receiving the first byte of data after the SPI chip selection is valid,, and the RB_SPI_IF_FST_BYTE bit of interrupt flag register will be set to 1, This bit is only valid in device mode.	0
3	RB_SPI_MST_SCK_MOD	RW	Host clock sampling mode configuration bits: 1: Mode 3 (SCK is at high level in idle	0

			mode); 0: Mode 0 (SCK is at low level in idle mode). This bit is only valid in host mode.	
2	RB_SPI_2WIRE_MOD	RW	2-wire or 3-wire SPI mode configuration bit, only for SPI0, SPI1 does not need this control bit: 1: 2-wire mode (SCK, MISO); 0: 3-wire mode (SCK, MOSI, MISO).	0
1	RB_SPI_ALL_CLEAR	RW	FIFO register and counter register clear bit: 1: Force to clear; 0: Not clear.	1
0	RB_SPI_MODE_SLAVE	RW	SPI0 master/slave mode selection bit, only for SPI0 1: Device mode; 0: Host mode. Note: SPI1 does not support device mode.	0

SPI configuration register (R8_SPIx_CTRL_CFG) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved.	0
5	RB_SPI_BIT_ORDER	RW	SPI data bit sequence selection bit: 1: Low byte is the first; 0: High byte is the first.	0
4	RB_SPI_AUTO_IF	RW	Enable the function of automatically clearing flag bit RB_SPI_IF_BYTE_END when accessing BUFFER/FIFO: 1: Enable; 0: Disable.	0
3	Reserved	RO	Reserved.	0
2	RB_SPI_DMA_LOOP	RW	SPI0 DMA address cycle enable bit: 1: Enable DMA address loop function; 0: Disable DMA address loop function. Note: Not supported by SPI1.	0
1	Reserved	RO	Reserved.	0
0	RB_SPI_DMA_ENABLE	RW	SPI0 DMA enable/disable bit, only supported by SPI0: 1: Enable DMA; 0: Disable DMA.	0

Note: If the DMA address cycle mode function is enabled, when the DMA address is added to the set end address, it will automatically cycle to the set first address, without resetting the DMA start address register (R16_SPI0_DMA_BEG) and DMA end address register (R16_SPI0_DMA_END).

SPI interrupt enable register (R8_SPIx_INTER_EN) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_SPI_IE_FST_BYTE	RW	In slave mode, the first byte receiving interrupt enable bit, only supported by	0

			SPI0: 1: Enable the first byte received to generate an interrupt; 0: Disable the first byte received to generate an interrupt. To enable this function, you need to set SPI to device mode, and meanwhile it is required to set the RB_SPI_SLV_CMD_MOD bit to 1, thus entering the first byte command mode.	
[6:5]	Reserved	RO	Reserved.	0
4	RB_SPI_IE_FIFO_OV	RW	FIFO overflow interrupt enable bit, only supported by SPI0: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	RB_SPI_IE_DMA_END	RW	DMA end interrupt enable bit, only supported by SPI0: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	RB_SPI_IE_FIFO_HF	RW	FIFO half use interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	RB_SPI_IE_BYTE_END	RW	SPI single byte transmission completion interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	RB_SPI_IE_CNT_END	RW	SPI all byte transmission completion interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

SPI master mode clock divider register (R8_SPIx_CLOCK_DIV) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPI_CLOCK_DIV	RW	Frequency division factor in master mode, the minimum value is 2. SPI clock frequency = main frequency/frequency division factor.	10h

Preset data register in SPI device mode (R8_SPI0_SLAVE_PRE)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPI0_SLAVE_PRE	RW	Preset return data in SPI0 device mode. Used to receive the return data after first byte of data. Note: Not supported by SPI1	10h

SPI data buffer (R8_SPIx_BUFFER) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPIx_BUFFER	RW	SPI data transmitting and receiving buffer	xx

SPI working status register (R8_SPIx_RUN_FLAG) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_SPI_SLV_SELECT	RO	SPI0 device mode selection status bit: 1: Device mode; 0: Host mode. Note: Not supported by SPI1	0
6	RB_SPI_SLV_CS_LOAD	RO	Status bit is loaded for the first time after SPI0 device mode is chip selected: 1: Loading is completed; 0: Not completed (the preload value can be modified). Note: Not supported by SPI1	0
5	RB_SPI_FIFO_READY	RO	FIFO ready status bit: 1: FIFO is ready; 0: FIFO is not ready.	0
4	RB_SPI_SLV_CMD_ACT	RO	Command receiving completion status bit in the SPI0 device mode, namely to completing the exchange of first byte data: 1: The first byte exchange is completed; 1: The first byte exchange is not completed. Note: Not supported by SPI1	0
[3:0]	Reserved	RO	Reserved.	0

SPI interrupt flag register (R8_SPIx_INT_FLAG) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_SPI_IF_FST_BYTE	RW1	First byte receiving flag bit in SPI0 device mode: 1: The first byte is received; 0: The first byte is not received. Note: Not supported by SPI1.	0
6	RB_SPI_FREE	RO	Current SPI idle state: 1: SPI is currently in idle state; 0: SPI is currently in a non-idle state.	0
5	Reserved	RO	Reserved.	0
4	RB_SPI_IF_FIFO_OV	RW1	SPI0 FIFO overflow flag bit, only supported by SPI0: 1: FIFO overflow; 0: No FIFO overflow. Note: Not supported by SPI1.	0
3	RB_SPI_IF_DMA_END	RW1	SPI0 DMA end flag bit, only supported by SPI0: 1: DMA transmission is ended; 0: DMA transmission is not ended. Note: Not supported by SPI1.	0
2	RB_SPI_IF_FIFO_HF	RW1	FIFO data half use flag bit: 1: Data reaches half of the FIFO buffer;	0

			0: Data does not reach half of the FIFO buffer; Note: RB_SPI_FIFO_DIR=1, receive data, trigger when FIFO count >= 4; RB_SPI_FIFO_DIR=0, send data, trigger when FIFO count <4.	
1	RB_SPI_IF_BYTE_END	RW1	SPI single byte transmission completion flag bit: 1: SPI single byte transmission is completed; 0: SPI single byte transmission is not completed.	0
0	RB_SPI_IF_CNT_END	RW1	SPI all bytes transmission completion flag bit: 1: SPI all bytes transmission is completed; 0: SPI all bytes transmission is not completed.	0

SPI transceiver FIFO counter register (R8_SPIx_FIFO_COUNT) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPIx_FIFO_COUNT	RW	Count byte in the current FIFO.	xx

SPI transceiver FIFO counter register (R8_SPIx_FIFO_COUNT1) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPIx_FIFO_COUNT1	RW	Count byte in the current FIFO, that is equivalent to the register R8_SPIx_FIFO_COUNT.	xx

SPI transceiver data total length register (R16_SPIx_TOTAL_CNT) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	R16_SPIx_TOTAL_CNT	RW	The total number of bytes of SPI data sent and received, and the lower 12 bits are valid. At most 4095 bytes can be transmitted at a time when using DMA.	0

SPI FIFO register (R8_SPIx_FIFO) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPIx_FIFO	RO/ WO	SPI FIFO register. The FIFO size is 8 bytes.	0

Register R8_SPIx_BUFFER and register R8_SPIx_FIFO are SPI data related registers. The main difference is: After the latter reads one byte of data, the value of length register (R16_SPI_TOTAL_CNT) is automatically reduced by 1 because it reads from FIFO. After the former reads one byte, the length register value remains unchanged.

Current address of SPI0 DMA buffer (R16_SPI0_DMA_NOW)

Bit	Name	Access	Description	Reset value

[15:0]	R16_SPI0_DMA_NOW	RW	Current address of DMA buffer, only supported by SPI0. The DMA operation can be judged by querying this value.	xxxx

Start address of SPI0 DMA buffer (R16_SPI0_DMA_BEG)

Bit	Name	Access	Description	Reset value
[15:0]	R16_SPI0_DMA_BEG	RW	Start address of DMA buffer, only supported by SPI0. Point to the start address of SPI0 receiving and transmiting data buffer.	xxxx

End address of SPI0 DMA buffer (R16_SPI0_DMA_END)

Bit	Name	Access	Description	Reset value
[15:0]	R16_SPI0_DMA_END	RW	End address of DMA buffer, only supported by SPI0. Point to the end address of SPI0 receiving and transmiting data buffer.	xxxx

6.3 SPI Transmission Format

SPI supports two transmission formats of mode 0 and mode 3, which can be selected by setting RB_SPI_MST_SCK_MOD bit of SPI mode configuration register (R8_SPIx_CTRL_MOD). The data transmission format is shown in the figure below:

Mode 0: RB_SPI_MST_SCK_MOD = 0

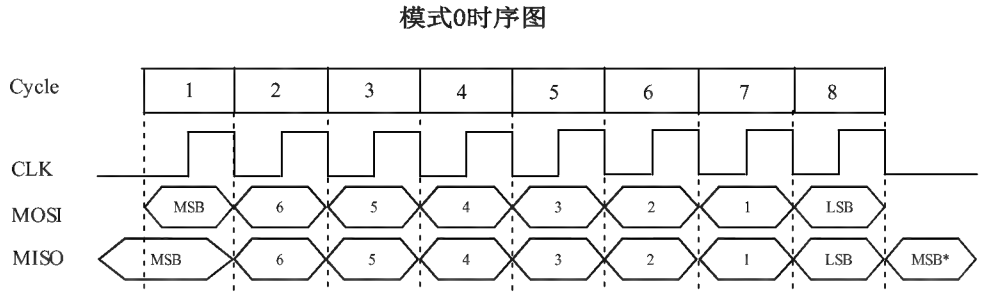


Figure 6-1 Transmission Format of SPI Mode 0

Mode 3: RB_SPI_MST_SCK_MOD = 1

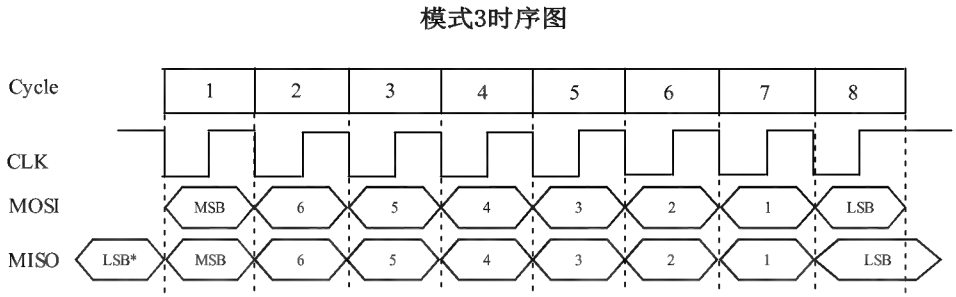


Figure 6-2 Transmission Format of SPI Mode 3

6.4 SPI Configuration

6.4.1 SPI Master Mode

In SPI master mode, serial clock is generated on SCK pin, and chip selection pins can be specified as any I/O pin.

Configuration Steps:
(1). Set SPI master mode clock divider register (R8_SPIx_CLOCK_DIV), to configure SPI clock speed;
(2). Set the RB_SPI_MODE_SLAVE bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 0, to configure SPI as the master mode;
(3). Set the RB_SPI_SLV_CMD_MOD bit of SPI mode configuration register (R8_SPIx_CTRL_MOD), and set it to mode 0 or mode 3 according to the requirements of the connected device;
(4). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD), and configure the FIFO direction. If it is 1, the current FIFO direction is data input. If it is 0, the current FIFO direction is data output.
(5). Set the RB_SPI_MOSI_OE bit and RB_SPI_SCK_OE bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 1, and set the RB_SPI_MISO_OE bit to 0, and set the bits corresponding to the MOSI pin and SCK pin in the PA port direction register (R32_PB_DIR) to 1, set the bit corresponding to MISO pin to 0, to configure the MOSI pin and SCK pin direction as output, and configure the MISO pin direction as input;

Data sending process:
(1). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 0, and configure the current FIFO direction as output;
(2). Write R16_SPIx_TOTAL_CNT register, and set the length of the data to be sent;
(3). Write R8_SPIx_FIFO register, write the data to be sent to FIFO. If R8_SPI0_FIFO_COUNT is less than FIFO size, data can continue to be written to FIFO;
(4). After all data is written to FIFO, wait until R16_SPIx_TOTAL_CNT register becomes 0, indicating that the data transmission is completed. If only one byte is to sent, you can also wait until R8_SPI0_FIFO_COUNT becomes 0, indicating that there is no data in FIFO and the data transmission is completed.

Data receiving process:
(1). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 1, to configure the current FIFO direction as input;
(2). Write R16_SPIx_TOTAL_CNT register, and set the length of the data to be sent;
(3). Wait until R8_SPIx_FIFO_COUNT register is not 0, indicating that the returned data has been received;
(4). The value read in R8_SPIx_FIFO is the received data.

6.4.2 SPI Slave Mode

Only SPI0 supports the slave mode. In the slave mode, SCK pin is used to receive the serial clock of the connected SPI host.

Configuration Steps:
(1). Set the RB_SPI_MODE_SLAVE bit of SPI0 mode configuration register (R8_SPI0_CTRL_MOD) to 1, to configure SPI0 as the slave mode;
(2). Set the RB_SPI_SLV_CMD_MOD bit of SPI0 mode configuration register (R8_SPI0_CTRL_MOD) as required;
(3). Set the RB_SPI_FIFO_DIR bit of SPI0 mode configuration register (R8_SPI0_CTRL_MOD), to configure the FIFO direction. If it is 1, the current FIFO direction is data input. If it is 0, the current FIFO direction is data output.

(4). Set the RB_SPI_MOSI_OE bit and RB_SPI_SCK_OE bit of SPI0 mode configuration register (R8_SPI0_CTRL_MOD) to 0, and set the RB_SPI_MISO_OE bit to 1, and set the bits corresponding to the MOSI pin, SCK pin and SCS pin in the PA port direction register (R32_PB_DIR) to 0, set the bit corresponding to MISO pin to 1, to configure the MOSI pin, SCK pin and SCS pin direction as input, and configure the MISO pin direction as output. If the MISO pin is not configured as output, when SCS chip selection is valid (low level), MISO will automatically enable output. It is recommended to set the MISO pin as input so that MISO does not output when chip selection is invalid, so that SPI bus can be shared when multiple devices operates. Note: For I/O pin direction of MISO in SPI slave mode, in addition to setting it as output, it also can be automatically configured as output during the period of valid SPI chip selection, but its output data is selected by RB_SPI_MISO_OE. When RB_SPI_MISO_OE is 1, SPI data is output. When RB_SPI_MISO_OE is 0, GPIO register data is output.
(5). Optionally, set the preset data register in SPI0 device mode (R8_SPI0_SLAVE_PRE), to be automatically loaded into the buffer for the first time after chip selection for external output. After 8 clocks (that is, the exchange of the first byte data between host and slave is completed), the controller will obtain the first byte data (command code) from the external SPI host, and the external SPI host exchanges and the preset data (status value) in R8_SPI0_SLAVE_PRE can be obtained. The bit 7 of this register will be automatically loaded into the MISO pins during the low level period of SCK after valid SPI chip selection. For SPI mode 0 (CLK defaults to low level), if the bit 7 of R8_SPI0_SLAVE_PRE is preset, the external SPI master will obtain the preset value of bit 7 of R8_SPI0_SLAVE_PRE by inquiring the MISO pins when the SPI chip selection is valid but there is no data transmission, thereby the value of bit 7 of R8_SPI0_SLAVE_PRE can be obtained only by the valid SPI chip selection.

Data sending process:
(1). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 0, to configure FIFO direction as data output;
(2) Write the data sent into the FIFO register (R8_SPIx_FIFO), and add 1 to the SPI transmiting/receiving data total length register (R16_SPIx_TOTAL_CNT). It is recommended to set R16_SPIx_TOTAL_CNT to a larger value at one time. In this way, the data in FIFO will be sent automatically, and it will be automatically suspended when the FIFO is empty. It is not necessary to set R16_SPIx_TOTAL_CNT every time;
(3). If a single byte is sent, wait for the R16_SPIx_TOTAL_CNT register to be 0, and wait for the data to be sent. If multiple bytes are to sent, you can write up to 8 data to the FIFO register (R8_SPIx_FIFO) at a time, and then wait for completing transmiting.

Data receiving process:
(1). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 1, to configure the current FIFO direction as data input;
(2) Wait to query the SPI transmiting/receiving data total length register (R16_SPIx_TOTAL_CNT). If the register is not 0, data is received, and the received data can be obtained by reading the FIFO register (R8_SPIx_FIFO).

For receiving the data of single byte, FIFO is not required, and SPI data buffer register (R8_SPIx_BUFFER) can be read directly to obtain the current data given by the other party for starting transmission.

6.5 DMA Function

For CH568, only SPI0 has DMA function, but SPI1 does not have this function. By enabling the DMA function, the receiving and transmiting of SPI data can be realized more conveniently on the basis of reducing software intervention.

6.5.1 DMA Sending Data Configuration in SPI Master Mode

(1). According to Section 6.4.1, configure SPI0 as master mode;
(2). If you need to generate DMA to complete interrupt, set the RB_SPI_IE_DMA_END bit of SPI interrupt enable register (R8_SPIx_INTER_EN) to 1;
(3). Initialize the R16_SPI_DMA_BEG register as the start address of SPI data sending buffer;
(4). Initialize the R16_SPI_DMA_END register as the end address of SPI data sending buffer;
(5). Clear SPI interrupt status register (R8_SPIx_INT_FLAG);
(6). Initialize the R16_SPI_TOTAL_CNT register to the number of data to be sent, and start the transmission if DMA is enabled;
(7). If you need to enable the DMA address loop mode function, you need to set the RB_SPI_DMA_LOOP bit of SPI DMA control register (R8_SPIx_CTRL_DMA) to 1;
(8). Set the RB_SPI_DMA_ENABLE bit of DMA control register (R8_SPIx_CTRL_DMA) of SPI to 1, and enable DMA to transmit data.

6.5.2 DMA Receiving Data Configuration in SPI Master Mode

(1). According to Section 6.4.1, configure SPI0 as master mode;
(2). If you need to generate a DMA interrupt, set the RB_SPI_IE_DMA_END bit of register R8_SPI_INTER_EN to 1, to enable the DMA end interrupt to be generated;
(3). Initialize the R16_SPI_DMA_BEG register as the start address of SPI data receive buffer;
(4). Initialize the R16_SPI_DMA_END register as the end address of SPI data receive buffer;
(5). Clear SPI interrupt status register (R8_SPIx_INT_FLAG);
(6). Initialize the R16_SPI_TOTAL_CNT register to the number of data to be received, and start the transmission if DMA is enabled;
(7). If you need to enable the DMA address loop mode function, you need to set the RB_SPI_DMA_LOOP bit of SPI DMA control register (R8_SPIx_CTRL_DMA) to 1;
(8). Set the RB_SPI_DMA_ENABLE bit of DMA control register (R8_SPIx_CTRL_DMA) of SPI to 1, and enable DMA to receive data. If DMA is enabled first, transmission will start automatically after setting R16_SPI_TOTAL_CNT.

Chapter 7 Universal Asynchronous Receiver/Transmitter (UART)

7.1 Introduction to UART

CH568 is equipped with 4 sets of full-duplex UARTs, UART0/1/2/3. Both full-duplex and half-duplex serial port communication are supported. Among them, UART0 is equipped with the transmiting status pin for switching RS485, and supports MODEM signals CTS, DSR, RI, DCD, DTR and RTS.

UART features:
(1). Compatible with 16C550 asynchronous serial port and enhanced;
(2). Support 5, 6, 7 or 8 data bits and 1 or 2 stop bits;
(3). Support the parity check modes of odd, even, no check, blank 0 and flag 1;
(4). Programmable communication baud rate, support 115200bps and communication baud rate up to 6Mbps;
(5). Built-in 8-byte FIFO buffer, support 4 FIFO trigger levels;
(6). UART0 supports MODEM signals CTS, DSR, RI, DCD, DTR and RTS, which can be converted to RS232 level;
(7). Support automatic handshake and automatic transmission rate control of hardware flow control signals CTS and RTS, compatible with TL16C550C;
(8). Support serial port frame error detection and Break line interval detection;
(9). Support full-duplex and half-duplex serial communication, and UART0 provides a transmiting status pin for switching RS485;

7.2 Register Description

UART0 related register physical start address: 0x0040 3000
UART1 related register physical start address: 0x0040 3400
UART2 related register physical start address: 0x0040 3800
UART3 related register physical start address: 0x0040 3c00

Table 7-1 List of UART0 Related Registers

Name	Offset address	Description	Reset value
R8_UART0_MCR	0x00	MODEM control register	8h00
R8_UART0_IER	0x01	Interrupt enable register	8h00
R8_UART0_FCR	0x02	FIFO control register	8h00
R8_UART0_LCR	0x03	Line control register	8h00
R8_UART0_IIR	0x04	Interrupt recognition register	8h01
R8_UART0_LSR	0x05	Line status register	8hC0
R8_UART0_MSR	0x06	MODEM status register	8hx0
R8_UART0_RBR	0x08	Receiver buffer register	8hxx
R8_UART0_THR	0x08	Transmitter hold register	8hxx
R8_UART0_RFC	0x0A	Receiver FIFO counter register	8hxx
R8_UART0_TFC	0x0B	Transmitter FIFO counter register	8hxx
R16_UART0_DL	0x0C	Baud rate divisor latch	16hxxxx
R8_UART0_DIV	0x0E	Predivider divisor register	8hxx
R8_UART0_ADR	0x0F	Slave address register	8hFF

Table 7-2 List of UART1 Related Registers

Name	Offset address	Description	Reset value
R8_UART1_MCR	0x00	MODEM control register	8h00
R8_UART1_IER	0x01	Interrupt enable register	8h00
R8_UART1_FCR	0x02	FIFO control register	8h00
R8_UART1_LCR	0x03	Line control register	8h00
R8_UART1_IIR	0x04	Interrupt recognition register	8h01
R8_UART1_LSR	0x05	Line status register	8hC0
R8_UART1_RBR	0x08	Receiver buffer register	8hxx
R8_UART1_THR	0x08	Transmitter hold register	8hxx
R8_UART1_RFC	0x0A	Receiver FIFO counter register	8hxx
R8_UART1_TFC	0x0B	Transmitter FIFO counter register	8hxx
R16_UART1_DL	0x0C	Baud rate divisor latch	16hxxxx
R8_UART1_DIV	0x0E	Predivider divisor register	8hxx

Table 7-3 List of UART2 Related Registers

Name	Offset address	Description	Reset value
R8_UART2_MCR	0x00	MODEM control register	8h00
R8_UART2_IER	0x01	Interrupt enable register	8h00
R8_UART2_FCR	0x02	FIFO control register	8h00
R8_UART2_LCR	0x03	Line control register	8h00
R8_UART2_IIR	0x04	Interrupt recognition register	8h01
R8_UART2_LSR	0x05	Line status register	8hC0
R8_UART2_RBR	0x08	Receiver buffer register	8hxx
R8_UART2_THR	0x08	Transmitter hold register	8hxx
R8_UART2_RFC	0x0A	Receiver FIFO counter register	8hxx
R8_UART2_TFC	0x0B	Transmitter FIFO counter register	8hxx
R16_UART2_DL	0x0C	Baud rate divisor latch	16hxxxx
R8_UART2_DIV	0x0E	Predivider divisor register	8hxx

Table 7-4 List of UART3 Related Registers

Name	Offset address	Description	Reset value
R8_UART3_MCR	0x00	MODEM control register	8h00
R8_UART3_IER	0x01	Interrupt enable register	8h00
R8_UART3_FCR	0x02	FIFO control register	8h00
R8_UART3_LCR	0x03	Line control register	8h00
R8_UART3_IIR	0x04	Interrupt recognition register	8h01
R8_UART3_LSR	0x05	Line status register	8hC0
R8_UART3_RBR	0x08	Receiver buffer register	8hxx
R8_UART3_THR	0x08	Transmitter hold register	8hxx
R8_UART3_RFC	0x0A	Receiver FIFO counter register	8hxx
R8_UART3_TFC	0x0B	Transmitter FIFO counter register	8hxx
R16_UART3_DL	0x0C	Baud rate divisor latch	16hxxxx
R8_UART3_DIV	0x0E	Predivider divisor register	8hxx

MODEM control register (R8_UARTx_MCR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
7	RB_MCR_HALF	RW	UART0 half-duplex transceiver mode	0

			enable/disable bit 1: Enter half-duplex transceiver mode, giving priority to transmission, receiving when not transmitting; 0: Disable half-duplex transceiver mode. Note: This only supports UART0.	
6	RB_MCR_TNOW	RW	Being transmiting status (TNOW) output (DTR pin) enable bit of UART0: 1: Enable the DTR pin of UART0 to output status of being transmiting TNOW, which can be used to control switch of RS485 transceiving; 0: Disable. Note: This only supports UART0.	0
5	RB_MCR_AU_FLOW_EN	RW	UART0 allow CTS and RTS hardware automatic flow control bit: 1: Allow CTS and RTS hardware automatic flow control; 0: Invalid. Note: This only supports UART0. In the flow control mode, if this bit is 1, then UART will continuously send the next data only when it detects that the CTS pin input is valid (active low). Otherwise, the serial port transmission will be suspended. And the change of CTS input status will not generate MODEM status Interrupt when this bit is 1. If this bit is 1 and RTS is 1, UART will automatically validate the RTS pin (active low) when receiver FIFO is empty. And UART will automatically invalidate the RTS pin until the number of received bytes reaches the trigger point of FIFO and can re-validate RTS pin when receiver FIFO is empty. The CTS pin can be connected to the other party RTS pin through hardware automatic band rate control, and the RTS pin to the other CTS pin.	0
4	RB_MCR_LOOP	RW	Test mode control bit of UART0 internal loop: 1: Enable the test mode of internal loop; 0: Disable the test mode of internal loop. In the test mode of the internal loop, all external output pins of UART are invalid, TXD internally returns to RXD (i.e., the output of TSR internally returns to the input of RSR), RTS internally returns to CTS, DTR internally returns to DSR,	0

			OUT1 internal returns to RI and OUT2 internally returns to DCD Note: This only supports UART0.	
3	RB_MCR_OUT2	RW	UART interrupt request output control bit: 1: Enable; 0: Disable.	0
2	RB_MCR_OUT1	RW	User-defined MODEM control bit, and no actual output pin connected: 1: Set to high level; 0: Set to low level. Note: This only supports UART0.	0
1	RB_MCR_RTS	RW	RTS pin output valid control bit: 0: Enable RTS pin output valid (active low) 0: Disable RTS pin output valid Note: This only supports UART0.	0
0	RB_MCR_DTR	RW	DTR pin output valid control bit: 1: Enable pin output valid (active low); 0: Disable pin output valid. Note: This only supports UART0.	0

Interrupt enable register (R8_UARTx_IER) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
7	RB_IER_RESET	WZ	UART software reset control bit, cleared automatically: 1: The software resets the UART; 0: Do not perform the software reset.	0
6	RB_IER_TXD_EN	RW	UART TXD pin output enable bit: 1: Enable pin output; 0: Disable pin output.	0
5	RB_IER_RTS_EN	RW	RTS pin output enable bit of UART0: 1: Enable pin output; 0: Disable pin output. Note: This only supports UART0.	0
4	RB_IER_DTR_EN	RW	DTR/TNOW pin output enable bit of UART0: 1: Enable pin output; 0: Disable pin output. Note: This only supports UART0.	0
3	RB_IER_MODEM_CHG	RW	Modulator-demodulator input status change interrupt enable bit of UART0: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt. Note: This only supports UART0.	0
2	RB_IER_LINE_STAT	RW	Receiver line status interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	RB_IER_THR_EMPTY	RW	Transmitter hold register empty interrupt enable bit:	0

			1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	
0	RB_IER_RECV_RDY	RW	Receiver data interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

FIFO control register (R8_UARTx_FCR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:6]	RB_FCR_FIFO_TRIG	RW	Interrupt of receiver FIFO and hardware flow control set trigger point domain: 00: 1 byte; 01: 2 bytes; 10: 4 bytes; 11: 7 bytes. This domain is used to set the interrupt of receiver FIFO and trigger point of hardware flow control. For example: 00 corresponds to 1 byte, that is, interrupt available for receiving data is generated when 1 byte is received, and RTS pin is automatically disabled when hardware flow control is enabled.	0
[5:3]	Reserved	RO	Reserved	0
2	RB_FCR_TX_FIFO_CLR	WZ	Transmitter FIFO data clear enable bit, cleared automatically: 1: Clear the data of transmitter FIFO (excluding TSR); 0: Do not clear the data of transmitter FIFO.	0
1	RB_FCR_RX_FIFO_CLR	WZ	Receiver FIFO data clear enable bit, cleared automatically: 1: Clear the data of receiver FIFO (excluding RSR); 0: Do not clear the data of receiver FIFO.	0
0	RB_FCR_FIFO_EN	RW	FIFO enable bit: 1: Enable FIFO, internal FIFO size is 8 bytes; 0: Disable FIFO. After disabling FIFO, it is 16C450 compatible mode, which means that there is only one byte in FIFO (RECV_TG1=0, RECV_TG0=0, FIFO_EN=1), and it is recommended to enable FIFO.	0

Line control register (R8_UARTx_LCR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
7	RB_LCR_DLAB/ RB_LCR_GP_BIT	RW	UART common bit (user-defined).	0

6	RB_LCR_BREAK_EN	RW	Forced to generate BREAK line interval enable bit: 1: Forced to generate BREAK line interval. 0: Not generate BREAK line interval.	0
[5:4]	RB_LCR_PAR_MOD	RW	Parity check bit format setting domain 00: Odd check; 01: Even check; 10: Mark bit (MARK, set to 1); 11: Space bit (SPACE, cleared to 0). This domain is only valid when the RB_LCR_PAR_EN bit is 1.	0
3	RB_LCR_PAR_EN	RW	Parity bit enable bit: 1: Allow to generate parity bit when sending and check parity bit when receiving; 0: No parity bit.	0
2	RB_LCR_STOP_BIT	RW	Stop bit format setting bit: 0: One stop bit; 1: Two stop bits.	0
[1:0]	RB_LCR_WORD_SZ	RW	UART data length setting domain: 00: 5 data bits; 01: 6 data bits; 10: 7 data bits; 11: 8 data bits.	0

Interrupt identification register (R8_UARTx_IIR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:6]	RB_IIR_FIFO_ID	RO	UART FIFO enable status bit: 1: FIFO is enabled; 0: FIFO is not enabled.	0
[5:4]	Reserved	RO	Reserved.	0
[3:0]	RB_IIR_INT_MASK	RO	Interrupt flag domain: If the RB_IIR_NO_INT bit is 0, an interrupt is generated, and it needs to determine the interrupt source after reading this domain. See the table below for details	0
0	RB_IIR_NO_INT	RO	UART no interrupt flag bit: 1: No interrupt; 0: Interrupt.	1

The meaning of bit RB_IIR_NO_INT of interrupt recognition register (R8_UARTx_IIR) and each bit of RB_IIR_INT_MASK domain are shown in the following table:

Table 8-3 Meaning of RB_IIR_INT_MASK in IIR Register

IIR register bit				Priority	Interrupt Type	Interrupt sources	Method for clearing interrupt
IID3	IID2	IID1	NOINT				
0	0	0	1	None	No interrupt generated	No interrupt	

1	1	1	0	0	Bus address matching	The received 1 data is the serial bus address, and the address matches the preset slave value or the broadcast address. Note: This interrupt only applies to UART0.	Read IIR or disable multi-device mode
0	1	1	0	1	Receiving line status	OVER_ERR, PAR_ERR, FRAM_ERR, BREAK_ERR	Read LSR
0	1	0	0	2	Data receiving available	The number of bytes received reaches the trigger point of FIFO.	Read RBR
1	1	0	0	2	Data receiving timeout	No next data is received when the time of four data is exceeded.	Read RBR
0	0	1	0	3	THR register null	Transmitter hold register empty, or RB_IER_THR_EMPTY bit is changed from 0 to 1 for triggering.	Read IIR Or write THR
0	0	0	0	4	MODEM input change	Trigger by setting △CTS, △DSR, △RI and △DCD to 1.	Read MSR

Line status register (R8_UARTx_LSR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
7	RB_LSR_ERR_RX_FIFO	RO	Receiver FIFO error flag bit: 1: There is at least one PAR_ERR, FRAM_ERR or BREAK_ERR error in the receiver FIFO; 0: There is no error in receiver FIFO.	0
6	RB_LSR_TX_ALL_EMP	RO	Transmitter hold register (THR) and transmitter shift register (TSR) empty flag: 1: Both of them are empty; 0: Both of them are not empty.	1
5	RB_LSR_TX_FIFO_EMP	RO	Transmitter FIFO empty flag bit: 1: Transmitter FIFO is empty; 0: Transmitter FIFO is not empty.	1
4	RB_LSR_BREAK_ERR	RZ	BREAK line interval detection flag bit: 1: BREAK line interval is detected: 0: BREAK line interval is not detected:	0
3	RB_LSR_FRAME_ERR	RZ	Data frame error flag bit: 1: It means the frame error of the data being read from the receiver FIFO due to lack of a valid stop bit; 0: The data frame is correct.	0
2	RB_LSR_PAR_ERR	RZ	Parity error flag bit of received data: 1: It means that there is parity error of the data being read from the receiver FIFO; 0: Parity check is correct.	0
1	RB_LSR_OVER_ERR	RZ	Receiver FIFO buffer overflow flag bit: 1: Overflowed; 0: Not overflowed.	0
0	RB_LSR_DATA_RDY	RO	Receiver FIFO receive data flag bit: 1: There is data in FIFO;	0

			0: No data. After reading all the data in the FIFO, this bit will be automatically cleared.	

MODEM status register (R8_UART0_MSR) (only supported by UART0)

Bit	Name	Access	Description	Reset value
7	RB_MSR_DCD	RO	DCD pin status bit: 1: DCD pin is valid (active low); 0: DCD pin is invalid (high level). Note: This only supports UART0.	x
6	RB_MSR_RI	RO	RI pin status bit: 1: RI pin is valid (active low); 0: RI pin is invalid (high level). Note: This only supports UART0.	x
5	RB_MSR_DSR	RO	DSR pin status bit: 1: DSR pin is valid (active low); 0: DSR pin is invalid (high level). Note: This only supports UART0.	x
4	RB_MSR_CTS	RO	CTS pin status bit: 1: CTS pin is valid (active low); 0: CTS pin is invalid (high level). Note: This only supports UART0.	x
3	RB_MSR_DCD_CHG	RZ	DCD pin input status change flag bit: 1: Change; 0: No change. Note: This only supports UART0.	0
2	RB_MSR_RI_CHG	RZ	RI pin input status change flag bit: 1: Change; 0: No change. Note: This only supports UART0.	0
1	RB_MSR_DSR_CHG	RZ	DSR pin input status change flag bit: 1: Change; 0: No change. Note: This only supports UART0.	0
0	RB_MSR_CTS_CHG	RZ	CTS pin input status change flag bit: 1: Change; 0: No change. Note: This only supports UART0.	0

Receiver buffer register (R8_UARTx_RBR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_RBR	RO	Data receiver buffer register. If the DATA_RDY bit of LSR is 1, the received data can be read from this register; If FIFO_EN is 1, the data received from the serial port shift register RSR will be firstly stored in the receiver FIFO, and then read out through the register	x

Transmitter hold register (R8_UARTx_THR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_THR	WO	Transmitter hold register. Transmitter FIFO is included, used to write the data to be transmitted; if FIFOEN is 1, the written data will be firstly stored in the transmitter FIFO, and then output one by one through the transmitter shift register TSR.	x

Receiver FIFO counter register (R8_UARTx_RFC) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_RFC	RO	Data count in the current receiver FIFO. The maximum value is 8.	x

Transmitter FIFO counter register (R8_UARTx_TFC) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_TFC	RO	Data count in the current transmitter FIFO. The maximum value is 8.	x

Baud rate divisor latch (R16_UARTx_DL) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UARTx_DL	RW	16-bit divisor, used to calculate baud rate. Formula: Divisor = the internal reference clock of UART / 16 / the required communication baud rate. For example: If the internal reference clock of UART is 1.8432MHz and the required baud rate is 9600bps, then the divisor =1843200/16/9600=12.	x

Pre-frequency division divisor register (R8_UARTx_DIV) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_DIV	RW	Used to calculate the internal reference clock of UART, the lower 7 bits are valid. Formula: Divisor = Fsys*2 / internal reference clock of UART, the maximum value is 127. Example: If system main clock is 96MHz and the divisor is 104, then the internal reference clock of UART is 1.846MHz, and the difference between it and the commonly used reference clock 1.8432 is 0.16%	x

Slave address register (R8_UART0_ADR) , only used by UART0

Bit	Name	Access	Description	Reset value

[7:0]	R8_UART0_ADR	RW	UART0 slave address: FFh: Not used; Other: Slave address.	8hFF

R8_UART0_ADR presets the address of this computer as a slave, which is used to automatically compare the received addresses during multi-device communication, and generate an interrupt when the address matches or when the broadcast address 0FFH is received, meanwhile, it is allowed to receive subsequent data packets. It is not allowed to receive any data before the address does not match. After sending data or rewriting to the R8_UART0_ADR register, stop receiving any data, until the address matches again next time or the broadcast address is received.

When R8_UART0_ADR is 0FFH or RB_LCR_PAR_EN=0, the automatic comparison function of bus address is disabled.

When R8_UART0_ADR is not 0FFH and RB_LCR_PAR_EN=1, the automatic comparison function of bus address is enabled, and the following parameters should be configured: RB_LCR_WORD_SZ is 11b to select 8 data bits. For the case when the address byte is MARK (that is, the bit 9 of data byte is 0), RB_LCR_PAR_MOD should be set to 10b; for the case when the address byte is SPACE (that is, the bit 9 of data byte is 1), RB_LCR_PAR_MOD should be set to 11b.

7.3 UART Application

The UART0/1/2/3 output pins of CH568 chip are all 3.3V LVCMOS level. The pins in asynchronous serial mode include: data transmission pin (supported by UART0/1/2/3) and MODEM contact signal pin (only supported by UART0). Data transmission pins include: TXD pin and RXD pin, both of which are at high level by default. MODEM contact signal pins include: CTS pin, DSR pin, RI pin, DCD pin, DTR pin, RTS pin, all of which are at high level by default. All these MODEM communication signals can be used as general-purpose IO pins, controlled by the application program and their purposes can be defined.

4 sets of UARTs have built-in independent transceiver buffer and 8-byte FIFO, support simplex, half-duplex or full duplex asynchronous serial communication. Serial data includes 1 low-level start bit, 5, 6, 7 or 8 data bits, 0 or 1 additional verification code or flag bit, 1 or 2 high-level stop bits, and supports odd/even/mark/blank check. Support common communication baud rates: 1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K, 230.4K, 460.8K, 921.6K, 1.8432M, 2.7648M, 7.8125M, etc. The baud rate error of UART transmitting signal is less than 0.2%, and the allowable baud rate error of UART receiving signal is not greater than 2%.

7.3.1 Baud Rate Calculation

1) Calculate the reference clock, set the R8_UART0_DIV register, the maximum value is 127;
2) Calculate the baud rate and set R16_UART0_DL register;
Baud rate formula =Fsys * 2 / R8_UART0_DIV / 16 / R16_UART0_DL.

7.3.2 Serial Transmission

"THR register empty" interrupt sent by the serial port (the low 4 bits of IIR are 02H) refers to transmitter FIFO empty. The interrupt is cleared after IIR is read, or the interrupt may be cleared when the next data is written to THR. If only one byte is written to THR, it will soon generate a request of "transmitter hold register (THR) empty interrupt" as the byte is quickly transferred to the transmitter shift register (TSR) to start transmitting. At this time, the next data ready to be transmitted can be written. After all data in TSR is removed, serial transmission is completed, and RB_LSR_TX_ALL_EMP bit of LSR is 1 and valid.

In interrupt trigger mode, when it receives the THR empty interrupt, if FIFO is enabled, up to 8 bytes can be written to THR and FIFO at a time, then it will be transmitted automatically by the controller in sequence. If FIFO is disabled, only one byte can be written at a time. If no data needs to be transmitted, simply exit (the

interrupts have been automatically cleared when IIR is read earlier).

In the query mode, it can judge whether the transmitter FIFO is empty according to RB_LSR_TX_FIFO_EMP bit of LSR. If this bit is 1, it can write data to THR and FIFO. If FIFO is enabled, it can write up to 8 bytes at a time.

7.3.3 Serial Reception

Data availability interrupt received by the serial port (the low 4 bits of IIR are 04H) means that the number of existing data bytes in the receiver FIFO has reached or exceeded the FIFO trigger point set and selected by RB_FCR_FIFO_TRIG of FCR. The interrupt is cleared when the data is read from RBR to cause the number of bytes in the FIFO below the FIFO trigger point.

Data timeout interrupt received by the serial port (the low 4 bits of IIR are 0CH) means that there is at least one byte of data in the receiver FIFO, and the user has waited for the time equivalent to the time for receiving 4 data when the serial port receives data last time and the MCU takes the data last time. The interrupt is cleared when a new data is received again, or the interrupt can also be cleared when the MCU reads RBR once. When receiver FIFO is empty, RB_LSR_DATA_RDY bit of LSR is 0. When there is data in the receiver FIFO, it is valid when RB_LSR_DATA_RDY bit is 1.

In the interrupt trigger mode, you can read the R8_UARTx_RFC register to query the remaining data count in the current FIFO after receiving the interrupt that serial port receives data timeout, read all the data directly, or continuously query the RB_LSR_DATA_RDY bit of LSR. If this bit is valid, read the data until this bit becomes invalid. After receiving the interrupt for receiving data availability from serial port, you can read the number of bytes set by RB_FCR_FIFO_TRIG of FCR from RBR, and then directly read the data for the number of bytes, or you can read all the data in the current FIFO according to the RB_LSR_DATA_RDY bit and the R8_UARTx_RFC register.

In query mode, MCU can judge whether the receiver FIFO is empty according to the RB_LSR_DATA_RDY bit of LSR, or read the R8_UARTx_RFC register to get the data count in the current FIFO and get all the data received by the serial port.

7.3.4 Hardware Flow Control

Hardware flow control includes automatic CTS (RB_MCR_AU_FLOW_EN of MCR is 1) and automatic RTS (RB_MCR_AU_FLOW_EN and RB_MCR_RTS of MCR are 1).

If automatic CTS is enabled, CTS pin must be valid before the serial port sends data. The serial port transmitter detects CTS pin before sending the next data. When CTS pin state is valid, the transmitter sends the next data. In order to ensure that the transmitter stops sending the later data, CTS pin must be disabled before the intermediate moment of the last stop bit currently sent. The automatic CTS function reduces the interrupt applied to the MCU system. When hardware flow control is enabled, a change of CTS pin level does not trigger a MODEM interrupt as the controller automatically controls the transmitter based on CTS pin status. If automatic RTS is enabled, RTS pin output will be valid only when there is enough space in FIFO to receive data, and RTS pin output is disabled when the receiver FIFO is full. RTS pin output will be valid if all the data in the receiver FIFO is taken or cleared. When the trigger points for the receiver FIFO are reached (the number of existing bytes in the receiver FIFO is not less than the number of bytes set by RB_FCR_FIFO_TRIG of FCR), RTS pin output is invalid, and the transmitter of the other side is allowed to send another data after RTS pin is invalid. Once the data in the receiver FIFO is emptied, RTS pin will be automatically re-enabled, so that the transmitter of the other side restores sending. If both automatic CTS and automatic RTS are enabled (both RB_MCR_AU_FLOW_EN and RB_MCR_RTS of MCR register are 1), one side will not send data unless there is sufficient space in the receiver FIFO of the other side when RTS pin of one side is connected to CTS pin of the other side. Therefore, the hardware flow control can avoid FIFO overflow and timeout errors during serial port reception.

Chapter 8 General-purpose Timer (TMRx)

8.1 Introduction to TMRx

CH568 chip is equipped with 3 26-bit timers, TMR0, TMR1 and TMR2, and the longest timing interval is 2^26 clock cycles. All timers support capture, PWM and interrupt functions. And TMR1 and TMR2 support DMA functions.

Features:
(1). 3 26-bit timers, and the longest timing interval of each timer is 2^26 clock cycles;
(2). Each timer supports PWM function;
(3). Each timer supports capture function;
(4). Timer interrupt is supported by each timer, and among them TMR1 and TMR2 support DMA and interrupt;
(5). The capture function can be set to level change capture function and high or low level hold time capture function;
(6). PWM function supports dynamically adjust PWM duty cycle settings;

8.2 Register Description

TMR0 related register physical start address: 0x0040 2000
TMR1 related register physical start address: 0x0040 2400
TMR2 related register physical start address: 0x0040 2800

Table 8-1 List of TMR0 Related Registers

Name	Offset address	Description	Reset value
R8_TMR0_CTRL_MOD	0x00	Mode setting register	8h02
R8_TMR0_INTER_EN	0x02	Interrupt enable register	8h00
R8_TMR0_INT_FLAG	0x06	Interrupt flag register	8h00
R8_TMR0_FIFO_COUNT	0x07	FIFO counter register	8h00
R32_TMR0_COUNT	0x08	Current count value register	32h0000 0000
R32_TMR0_CNT_END	0x0C	Final count value setting register	32h0000 0000
R32_TMR0_FIFO	0x10	FIFO register	32h0000 0000

Table 8-2 List of TMR1 Related Registers

Name	Offset address	Description	Reset value
R8_TMR1_CTRL_MOD	0x00	Mode setting register	8h02
R8_TMR1_CTRL_DMA	0x01	DMA control register	8h00
R8_TMR1_INTER_EN	0x02	Interrupt enable register	8h00
R8_TMR1_INT_FLAG	0x06	Interrupt flag register	8h00
R8_TMR1_FIFO_COUNT	0x07	FIFO counter register	8h00
R32_TMR1_COUNT	0x08	Current count value register	32h0000 0000
R32_TMR1_CNT_END	0x0C	Final count value register	32h0000 0000
R32_TMR1_FIFO	0x10	FIFO register	32h0000 0000
R16_TMR1_DMA_NOW	0x14	Current address of DMA buffer	16h0000
R16_TMR1_DMA_BEG	0x18	Start address of DMA buffer	16h0000

R16_TMR1_DMA_END	0x1C	End address of DMA buffer	16h0000

Table 8-3 List of TMR2 Related Registers

Name	Offset address	Description	Reset value
R8_TMR2_CTRL_MOD	0x00	Mode setting register	8h02
R8_TMR2_CTRL_DMA	0x01	DMA control register	8h00
R8_TMR2_INTER_EN	0x02	Interrupt enable register	8h00
R8_TMR2_INT_FLAG	0x06	Interrupt flag register	8h00
R8_TMR2_FIFO_COUNT	0x07	FIFO counter register	8h00
R32_TMR2_COUNT	0x08	Current count value register	32h0000 0000
R32_TMR2_CNT_END	0x0C	Final count value register	32h0000 0000
R32_TMR2_FIFO	0x10	FIFO register	32h0000 0000
R16_TMR2_DMA_NOW	0x14	Current address of DMA buffer	16h0000
R16_TMR2_DMA_BEG	0x18	Start address of DMA buffer	16h0000
R16_TMR2_DMA_END	0x1C	End address of DMA buffer	16h0000

Mode setting register (R8_TMRx_CTRL_MOD) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:6]	RB_TMR_CAP_EDGE	RW	In capture mode, edge trigger mode setting domain: 00: Disable trigger; 01: Capture the time between any edge changes; 10: Capture the time between falling edges; 11: Capture the time between rising edges;	0
[7:6]	RB_TMR_PWM_REPEAT	RW	PWM repeat mode setting domain: 00: Repeat for 1 time; 01: Repeat for 4 times; 10: Repeat for 8 times; 11: Repeat for 16 times.	0
5	Reserved	RO	Reserved.	0
4	RB_TMR_CAP_COUNT	RW	Auxiliary selection, when RB_TMR_MODE_IN=1, set: 1: Count; 0: Capture.	0
4	RB_TMR_OUT_POLAR	RW	In PWM mode, output polarity setting bit: 1: Default high level, active low; 0: Default low level, active high;	0
3	RB_TMR_OUT_EN	RW	Timer output enable bit 1: Timer output is enabled; 0: Timer output is disabled.	0
2	RB_TMR_COUNT_EN	RW	Timer module enable bit: 1: Enable; 0: Disable.	0
1	RB_TMR_ALL_CLEAR	RW	FIFO and COUNT register of the counter clear bit: 1: Force to clear;	1

			0: No action.	
0	RB_TMR_MODE_IN	RW	Timer mode setting bit: 1: Capture/count mode 0: Timing mode/PWM mode	0

Interrupt enable register (R8_TMRx_INTER_EN) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	x
4	RB_TMR_IE_FIFO_OV	RW	FIFO overflow interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	RB_TMR_IE_DMA_END	RW	DMA end interrupt enable bit, not supported by TMR0: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	RB_TMR_IE_FIFO_HF	RW	FIFO half used interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt. (capture fifo >=4 or PWM fifo <=3)	0
1	RB_TMR_IE_DATA_ACT	RW	In capture mode, level change interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt. In PWM mode, PWM completion interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	RB_TMR_IE_CYC_END	RW	In capture mode, capture timeout interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt. In PWM mode, PWM clock cycle end interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

Interrupt flag register (R8_TMRx_INT_FLAG) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	x
4	RB_TMR_IF_FIFO_OV	RW1	FIFO overflow flag bit: 1: Overflowed; 0: No overflow.	0
3	RB_TMR_IF_DMA_END	RW1	DMA completion flag bit, not supported by TMR0: 1: Completed; 0: Not completed.	0
2	RB_TMR_IF_FIFO_HF	RW1	FIFO half count flag bit: 1: FIFO has been half counted;	0

			0: FIFO has not been half counted. (capture fifo >=4 or PWM fifo <=3)	
1	RB_TMR_IF_DATA_ACT	RW1	In capture mode, capture level change flag bit: 1: Capture the level change; 0: Level change is not captured. In PWM mode, PWM trigger flag bit: 1: Triggered (PWM count reaches the specified value); 0: Not triggered.	0
0	RB_TMR_IF_CYC_END	RW1	In capture mode, timeout flag bit: 1: Time out; 0: Not time out. In PWM mode, PWM cycle end flag bit: 1: End; 0: Not end. In timing mode: 1: Timing cycle ends; 0: Not end. Cleared by writing 1.	0

FIFO counter register (R8_TMRx_FIFO_COUNT) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:0]	R8_TMRx_FIFO_COUNT	RO	Data byte count in the FIFO, the maximum value is 8.	x

Current count value register (R32_TMRx_COUNT) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMRx_COUNT	RO	Current count value of counter.	x

Final counting value setting register (R32_TMRx_CNT_END) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMRx_CNT_END	RW	In timer mode, number of timing clocks; In PWM mode, the total number of PWM cycle clocks; In the capture mode, the number of captured timeout clock cycles; The maximum value is 67108864. Note: R32_TMRx_COUNT counts from 0, so the maximum value is R32_TMRx_CNT_END minus 1. Only the lower 26 bits are valid.	x

FIFO register (R32_TMRx_FIFO) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMRx_FIFO	RO/ WO	FIFO data register, only the lower 26 bits are valid.	x

DMA control register (R8_TMRx_CTRL_DMA) (x=1/2)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved.	x
2	RB_TMR_DMA_LOOP	RW	DMA address loop function enable bit, not supported by TMR0: 1: Enable DMA address loop function; 0: Disable DMA address loop function. If the DMA address loop function is enabled, when the DMA address is added to the set end address, it will automatically loop to the start address set.	0
1	Reserved	RO	Reserved.	0
0	RB_TMR_DMA_ENABLE	RW	DMA function enable bit, not supported by TMR0: 1: Enable DMA; 0: Disable DMA.	0

DMA current buffer address (R16_TMRx_DMA_NOW) (x=1/2)

Bit	Name	Access	Description	Reset value
[15:0]	R16_TMRx_DMA_NOW	RW	Current address of DMA data buffer. It can be used to calculate the number of conversions, and the calculation method is: COUNT=(TMR_DMA_NOW-TMR_DMA_BEG)/4	xxxx

DMA start buffer address (R16_TMRx_DMA_BEG) (x=1/2)

Bit	Name	Access	Description	Reset value
[15:0]	R16_TMRx_DMA_BEG	RW	Start address of DMA data buffer, 4 bytes must be aligned for the address. That is, in PWM data transmission or capture mode, the data captured starts from this buffer address.	xxxx

DMA end buffer address (16_TMRx_DMA_END) (x=1/2)

Bit	Name	Access	Description	Reset value
[15:0]	R16_TMRx_DMA_END	RW	End address of DMA data buffer, 4 bytes must be aligned for the address. That is, in PWM data transmission or capture mode, the data captured ends in this buffer address.	xxxx

8.3 TMRx Function

8.3.1 Timing and Counting Function

There are 3 timers in CH568, and the maximum timing interval of each timer is 2^26 clock cycles. If the system clock cycle is 96M, the longest time interval is: 10.4ns*2^26 ≈ 0.7s. If the system clock is lower than 96M, the time interval is longer.

Each timer has independent interrupt.

The timing function register is initialized as follows:
(1). Set register R32_TMRx_CNT_END to the timing value required;
The specific calculation method is: Time = Fsys * R32_TMRx_CNT_END
(2). Set the RB_TMR_MODE_IN bit of register R8_TMRx_CTRL_MOD to 0, and set RB_TMR_ALL_CLEAR to 0;
(3). Set the RB_TMR_COUNT_EN bit of register R8_TMRx_CTRL_MOD to 1, and start the timer function;
(4). At the end of timing interval, set the RB_TMR_IF_CYC_END bit of register R8_TMRx_INT_FLAG to 1, and cleared by writing 1.

8.3.2 PWM Function

3 timers of CH568 all have PWM function. The default output polarity of PWM can be set to high level or low level. The number of repeated times can be selected as 1, 4, 8 or 16. This repeat function is combined with DMA to imitate the effect of DAC. The shortest time cycle of PWM output is 1 system clock cycle, and the duty cycle of PWM can be dynamically modified to imitate special waveforms, such as quasi-sine-wave.

PWM function operation:

It is needed to set the register (R32_TMRx_FIFO) and register (R32_TMRx_CNT_END) when PWM outputs, R32_TMRx_FIFO as the data register, R32_TMRx_CNT_END as the PWM total cycle register.

PWM operation steps are as follows:
(1). Set the PWM total cycle register R32_TMRx_CNT_END, the minimum value is 1, the value of this register must be greater than or equal to the value of R32_TMRx_FIFO register;
(2). Set the data register R32_TMRx_FIFO, the minimum value is 0, with the corresponding duty cycle of 0%, the maximum value is the same as that of R32_TMR_CNT_END, with the corresponding duty cycle of 100%. The calculation of duty cycle: R32_TMRx_FIFO/R32_TMRx_CNT_END. TMR1 and TMR2 support continuous dynamic data (DMA), which can imitate special waveforms;
(3). Clear the RB_TMR_MODE_IN bit in the mode setting register (R8_TMRx_CTRL_MOD) to 0 to enable the PWM mode. At the same time, set the RB_TMR_ALL_CLEAR bit to 1 and then clear it to 0 forcefully to clear the FIFO and COUNT. Set the RB_TMR_OUT_POLAR bit to select the output polarity. If you need to set the number of repetitions, set the RB_TMR_PWM_REPEAT domain as needed.
(4). Set the RB_TMR_COUNT_EN and RB_TMR_OUT_EN bits in the mode setting register (R8_TMRx_CTRL_MOD) to 1 to enable the PWM function;
(5). Set the I/O pin corresponding to PWM as output;
(6). If you need to enable interrupts, set the corresponding interrupt enable register bit;
(7). After the PWM is completed, if the interrupt is turned on, the corresponding timer interrupt will be generated. At the same time, read the R8_TMRx_INT_FLAG register to know whether the PWM is completed and whether an error occurred during the PWM process;

For example: Set RB_TMR_OUT_POLAR bit to 0, R32_TMRx_FIFO to 6, R32_TMRx_CNT_END to 18, the generated basic timing diagram of PWM is as follows, and its duty cycle is:

PWM duty cycle = R32_TMRx_FIFO/R32_TMRx_CNT_END = 1/3

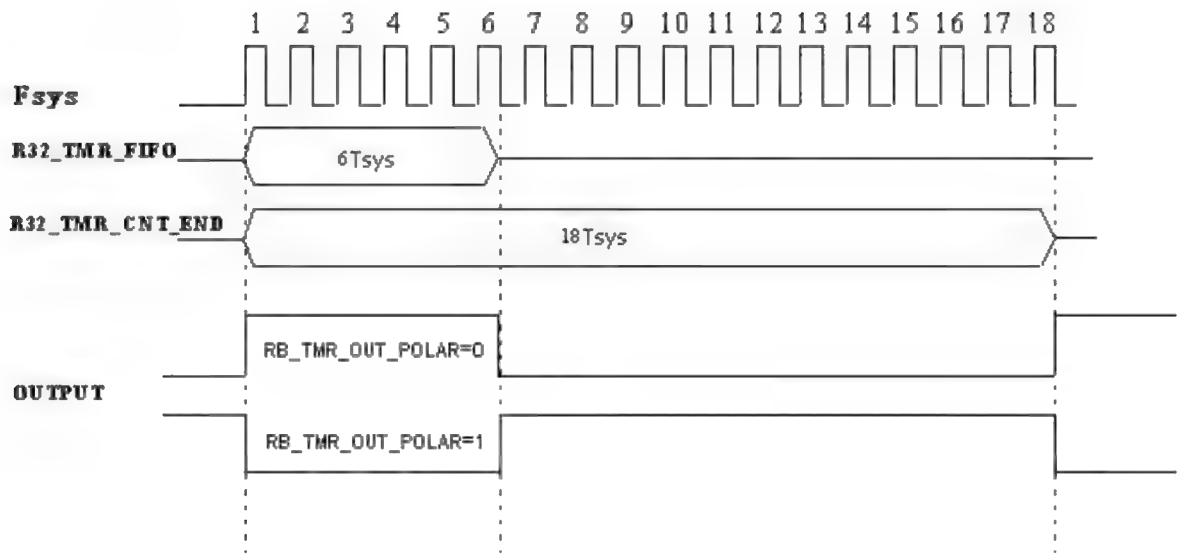


Figure 8-1 PWM Output Timing Diagram

If RB_TMR_PWM_REPEAT domain is set to 00, it means that the above process repeates once, 01 means that the above process repeates for 4 times, 10 means that the above process repeates for 8 times, and 11 means that the above process repeates for 16 times. After repeating, take the next data in FIFO and then continue.

8.3.3 Capture Function

3 timers of CH568 all have capture function, among which the capture functions of TMR1 and TMR2 support DMA data storage. Three capture modes can be selected: start triggering at any edge and end at any edge, start triggering at rising edge and end at rising edge, and start triggering at falling edge and end at falling edge The following is the description of capture trigger mode:

Table 8-4 Description of Capture Trigger Mode

Capture Mode Selection Bit (RB_TMR_CATCH_EDGE)	Trigger Mode	Graphical representation
00	Disable capture	None
01	Edge trigger	
10	Falling edge to falling edge	
11	Rising edge to rising edge	

There are two trigger states in edge trigger mode, which can capture high level width or low level width. When highest bit (bit 25) of the valid data in data register (R32_TMRx_FIFO) is 1, high level is captured; otherwise, low level is captured. If the bit 25 of multiple sets of data is 1 (or 0), the width of the high (or low) level exceeds the timeout value, and multiple sets need to be accumulated are required.

In the trigger modes from falling edge to falling edge or from rising edge to rising edge, an input change cycle can be captured. When the highest bit (bit 25) of the valid data in data register (R32_TMRx_FIFO) is 0, one cycle is normally sampled. When it is 1, the input change period exceeds the timeout value

R32_TMRx_CNT_END, and the latter set of data needs to be added and accumulated as a single input change cycle.

The specific description is shown in the figure below:

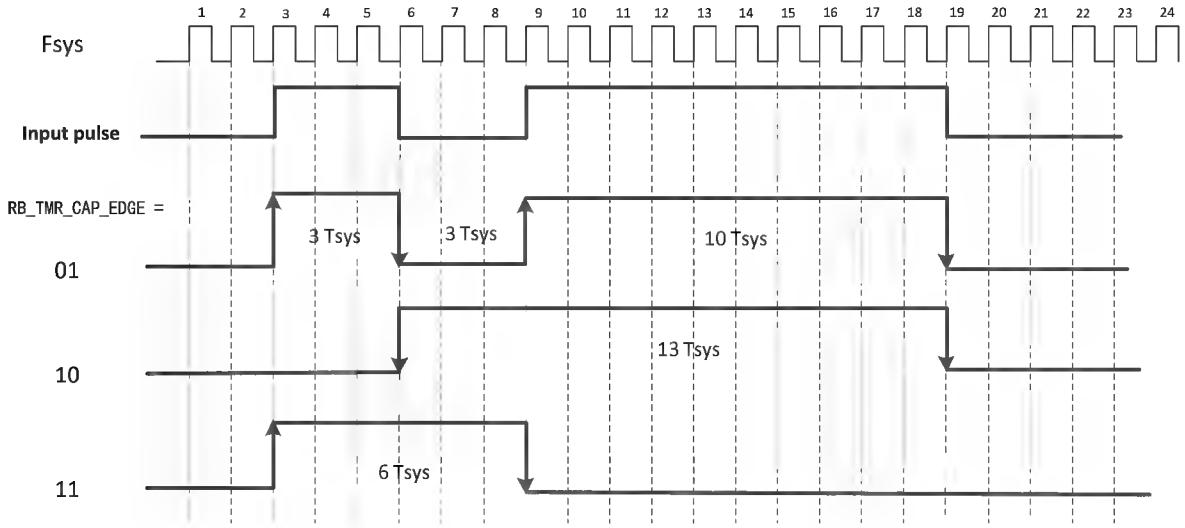


Figure 8-2 Taking System Clock Cycle as Capture Cycle

As shown in the figure above, sample once in each clock cycle,

When RB_TMR_CATCH_EDGE = 2b01, it is set as edge-triggered sampling, and the time widths sampled are 3Tsys, 3Tsys, 10Tsys;

When RB_TMR_CATCH_EDGE = 2b10, it is set as the sampling from falling edge to falling edge, and the time width sampled is 13Tsys;

When RB_TMR_CATCH_EDGE = 2b11, it is set as the sampling from rising edge to rising edge, and the time width sampled is 6Tsys.

Operation steps for capture mode:

(1). Set the register R32_TMRx_CNT_END to set the capture timeout time. The default maximum timeout time is 2^26 clock cycles. It is recommended to set a reasonable timeout to avoid no time data when the input is unchanged for a long time. If no level change is detected within the maximum timeout time, set bit 25 of R32_TMRx_FIFO register to 1;

(2). Set the direction of the I/O pin corresponding to capture as input;

(3). Set the RB_TMR_MODE_IN bit of mode setting register (R8_TMRx_CTRL_MOD) to 1 and RB_TMR_CAP_COUNT bit to 0, and reset the RB_TMR_ALL_CLEAR bit to clear FIFO and COUNT. At the same time, set the RB_TMR_CAP_EDGE to select the capture mode;

(4). If interrupt needs to be enabled, set the corresponding bit of interrupt register R8_TMRx_INTER_EN to 1, and enable the corresponding interrupt;

(5). To save the captured data by DMA method (only supported by TMR1 and TMR2), you need to set RB_TMR_DMA_ENABLE bit of R8_TMRx_CTRL_DMA register to 1, enable the DMA function, and set the register R16_TMRx_DMA_BEG to the first address of the buffer for storing the captured data, and set the register R16_TMRx_DMA_END as the end address of the buffer for storing captured data;

(6). Set the RB_TMR_COUNT_EN bit of register R8_TMRx_CTRL_MOD to 1, enable timer module and start the capture function;

(7). After the capture is completed, the register R8_TMRx_INT_FLAG will generate the corresponding interrupt status. The data captured by default is stored in the register R32_TMRx_FIFO. If DMA data transmission is used, the captured data will be automatically stored in the data buffer set by DMA.

Chapter 9 PWM

9.1 Introduction to PWM Controller

CH568 is provided with 4-channel PWM output, the duty cycle is adjustable, and the PWM cycle is fixed and 2 modes are available, and the operation is simple.

The extended PWM pin output is identified as PWM3/ PWM4/ PWM5/ PWM6, among which PWM5 and PWM6 support remapping to PWM5_ and PWM6_ pins.

9.2 Register Description

PWM0/1/2/3 related register physical base address: 0x0040 5000

Table 9-1 List of PWM0/1/2/3 Related Registers

Name	Offset address	Description	Reset value
R8_PWM_CTRL_MOD	0x00	PWM mode control register	8h00
R8_PWM_CTRL_CFG	0x01	PWM configuration control register	8h00
R8_PWM_CLOCK_DIV	0x02	PWM clock frequency division register	8h00
R32_PWM_DATA	0x08	PWM0/1/2/3 data hold register	32h

PWM mode control register (R8_PWM_CTRL_MOD)

Bit	Name	Access	Description	Reset value
7	RB_PWM3_POLAR	RW	PWM3 output polarity control bit: 1: Default high level, active low; 0: Default low level, active high;	0
6	RB_PWM2_POLAR	RW	PWM2 output polarity control bit: 1: Default high level, active low; 0: Default low level, active high;	0
5	RB_PWM1_POLAR	RW	PWM1 output polarity control bit: 1: Default high level, active low; 0: Default low level, active high;	0
4	RB_PWM0_POLAR	RW	PWM0 output polarity control bit: 1: Default high level, active low; 0: Default low level, active high;	0
3	RB_PWM3_OUT_EN	RW	PWM3 output enable bit: 1: Enable; 0: Disable.	0
2	RB_PWM2_OUT_EN	RW	PWM2 output enable bit: 1: Enable; 0: Disable.	0
1	RB_PWM1_OUT_EN	RW	PWM1 output enable bit: 1: Enable; 0: Disable.	0
0	RB_PWM0_OUT_EN	RW	PWM0 output enable bit: 1: Enable; 0: Disable.	0

PWM configuration control register (R8_PWM_CTRL_CFG)

Bit	Name	Access	Description	Reset value
[7:1]	Reserved	RW	Reserved.	0
0	RB_PWM_CYCLE_SEL	RW	PWM cycle selection: 1: 255 clock cycles; 0: 256 clock cycles.	0

PWM clock divider register (R8_PWM_CLOCK_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_PWM_CLOCK_DIV	RW	PWM reference clock frequency division factor. Calculation: CLK=Fsys/R8_PWM_CLOCK_DIV.	0

PWM0/1/2/3 data hold register (R32_PWM_DATA)

Bit	Name	Access	Description	Reset value
[31:24]	R8_PWM3_DATA	RW	PWM3 data hold register.	xx
[23:16]	R8_PWM2_DATA	RW	PWM2 data hold register.	xx
[15:8]	R8_PWM1_DATA	RW	PWM1 data hold register.	xx
[7:0]	R8_PWM0_DATA	RW	PWM0 data hold register.	xx

9.3 PWM Configuration

1) Set PWM0-PWM3 pin direction as output; optionally, set the drive capability of corresponding I/O;
2) Set the register R8_PWM_CLOCK_DIV to calculate the clock reference frequency of PWM;
3) Set the PWM mode control register R8_PWM_CTRL_MOD, configure the output polarity of PWMx, and enable the corresponding PWMx (RB_PWMx_OUT_EN position 1) output;
4) Set the R8_PWM_CTRL_CFG register and R32_PWM_DATA register to configure the PWM duty cycle output.

Calculation formula:

PWMx duty cycle = R8_PWMx_DATA / (RB_PWM_CYCLE_SEL? 255 : 256)

Note: If the corresponding RB_PWMx_OUT_EN output enable is always on in the R8_PWM_CTRL_MOD register, the PWM waveform will be output continuously until RB_PWMx_OUT_EN is disabled.

Chapter 10 LED Screen Controller

10.1 Introduction to LED Controller

CH568 is equipped with an LED screen control card interface and built-in 4-byte FIFO, which supports DMA and interrupts, saves CPU processing time and supports 1/2/4 data line interfaces.

10.2 Register Description

LED related register physical base address: 0x0040 6000

Table 10-1 List of LED Related Registers

Name	Offset address	Description	Reset value
R8_LED_CTRL_MOD	0x00	LED mode configuration register	8h02
R8_LED_CLOCK_DIV	0x01	LED serial clock divider register	8hxx
R8_LED_STATUS	0x04	LED status register	8h00
R16_LED_FIFO	0x08	LED FIFO register	16hxxxx
R16_LED_DMA_CNT	0x10	LED DMA remaining counter register	16hxxxx
R16_LED_DMA_MAIN	0x14	LED main buffer DMA address	16hxxxx
R16_LED_DMA_AUX	0x18	LED auxiliary buffer DMA address	16hxxxx

LED mode configuration register (R8_LED_CTRL_MOD)

Bit	Name	Access	Description	Reset value
[7:6]	RB_LED_CHAN_MOD	RW	LED channel mode setting domain: 00: LED0, single channel output; 01: LED0/1, 2-channel output; 10: LED0~3, 4-channel output; 11: LED0~3, 4-channel output, among which the data of LED2/3 channels is from auxiliary buffer.	0
5	RB_LED_IE_FIFO	RW	FIFO half count interrupt enable: 1: FIFO count<=2 interrupt trigger; 0: Disable the corresponding interrupt.	0
4	RB_LED_DMA_EN	RW	LED DMA function and DMA interrupt enable: 1: Enable; 0: Disable.	0
3	RB_LED_OUT_EN	RW	LED signal output: 1: Enable; 0: Disable.	0
2	RB_LED_OUT_POLAR	RW	LED data output polarity control bit: 1: Flip output, data 0 outputs 1, data 1 outputs 0; 0: Direct output, data 0 outputs 0, data 1 outputs 1.	0
1	RB_LED_ALL_CLEAR	RW	Clear LED FIFO and counter: 1: Force to clear;	1

			0: No action.	
0	RB_LED_BIT_ORDER	RW	LED serial data bit sequence: 1: High byte is the first; 0: Low byte is the first;	0

LED serial clock divider register (R8_LED_CLOCK_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_LED_CLOCK_DIV	RW	LED control output clock frequency division factor. Calculation: CLK=Fsys/R8_LED_CLOCK_DIV.	xx

LED status register (R8_LED_STATUS)

Bit	Name	Access	Description	Reset value
7	RB_LED_IF_DMA_END	RW1	DMA transmission completion flag bit: 1: DMA transmission is completed; 0: Not completed. Cleared by writing 1 or R16_LED_DMA_CNT	0
6	RB_LED_FIFO_EMPTY	RO	FIFO empty status bit: 1: FIFO is empty; 0: FIFO is not empty.	0
5	RB_LED_IF_FIFO	RW1	FIFO half count interrupt flag bit: 1: FIFO count <= 2; 0: FIFO count > 2. Cleared by writing 1 or R16_LED_FIFO	0
4	RB_LED_CLOCK	O	Current LED clock signal level status: 1: High level; 0: Low level.	0
3	Reserved	RO	Reserved.	0
[2:0]	RB_LED_FIFO_COUNT	RO	Byte count in current FIFO, must be an even number.	0

LED FIFO register (R16_LED_FIFO)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_FIFO	WO	LED data FIFO entry, 16-bit write.	xxxx

LED DMA remaining counter register (R16_LED_DMA_CNT)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_DMA_CNT	RW	Current DMA remaining word (16-bit) count of LED_DMA_MAIN main buffer area. It will automatically decrease after DMA is started and only the lower 12 bits are valid. Auxiliary buffers are not included.	xxxx

LED main buffer DMA address (R16_LED_DMA_MAIN)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_DMA_MAIN	RW	The DMA start address/current address of the main buffer area, automatically decreases after the initial value is preset.	xxxx

LED auxiliary buffer DMA address (R16_LED_DMA_AUX)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_DMA_AUX	RW	DMA start address/current address of auxiliary buffer area, automatically increases after the initial value is preset.	xxxx

10.3 LED Control Application

1) Set LEDC and the necessary LED0～LED3 pin directions as output, optionally, set the corresponding I/O drive capability;
2) Set R8_LED_CLOCK_DIV to select the LED output clock frequency;
3) Set the DMA start address R16_LED_DMA_MAIN to point to the buffer that is ready to output data, that is, the main buffer;
4) If LED channel mode 3 is selected, the auxiliary DMA starting address R16_LED_DMA_AUX must be set to point to the auxiliary buffer;
5) Set the LED control register R8_LED_CTRL_MOD, select the channel mode, output polarity, bit sequence, enable interrupt and DMA functions, etc.;
6) Set the DMA counter register R16_LED_DMA_CNT, start DMA transmission, or send data by writing to FIFO;
7) Query or use interrupt processing to interrupt the corresponding status.

Chapter 11 USB Controller

11.1 Introduction to USB Controller

CH568 is embedded with USB2.0 controller and USB-PHY, with dual roles of host controller and USB device controller. When used as a host controller, it can support low-speed, full-speed and high-speed USB devices. When used as a device controller, it can be flexibly set to low-speed, full-speed or high-speed mode, to adapt to various applications.

The features of USB controller are as follows:
1) Support USB 2.0, USB 1.1 and USB 1.0;
2) Support USB Host functions and USB Device functions;
3) Host supports high-speed HUB;
4) The hardware can be configured as high-speed, full-speed and low-speed device;
5) Both the host and the device support control transmission, bulk transmission, interrupt transmission, isochronous/synchronous transmission;
6) Support directly access the data of each endpoint buffer by DMA;
7) Support suspend, remote wake-up and resume functions;
8) Except device endpoint 0, all other endpoints support the data packets up to 512 bytes, and some endpoints support double buffering.

11.2 Register Description

CH568 is integrated with USB2.0 master/slave controller (built-in PHY) and it can be flexibly configured as a host function or device function.

The USB related registers of CH568 are divided into 3 parts, some of which are reused in the host and device modes.
1) USB global registers;
2) USB device controller registers;
3) USB host controller registers.

USB related register physical base address: 0x0040 9000

11.2.1 Global Register Description

Table 11-1 List of USB Global Registers

Name	Offset address	Description	Reset value
USB_CTRL	00h	USB control register	8h06
USB_INT_EN	02h	USB interrupt enable register	8h00
USB_DEV_AD	03h	USB device address register	8h00
USB_FRAME_NO	04h	USB frame number register	16h0000
USB_SUSPEND	06h	USB suspend control register	8h00
USB_SPPED_TYPE	08h	USB current speed type register	8h00
USB_MIS_ST	09h	USB miscellaneous status register	8hxx10_1000b
USB_INT_FG	0Ah	USB interrupt flag register	8h00
USB_INT_ST	0Bh	USB interrupt status register	8h00xx_xxxxb
USB_RX_LEN	0Ch	USB receiving length register	16hxxxx

USB control register (USB_CTRL)

Bit	Name	Access	Description	Reset value
7	bUC_HOST_MODE	RW	USB working mode selection bit: 0: Device mode (DEVICE); 1: Host mode (HOST).	0
[6:5]	UC_SPEED_TYPE	RW	USB bus signal transmission rate selection bit: 00: Full speed; 01: High speed; 10: Low speed.	00b
4	bUC_DEV_PU_EN	RW	In device mode, USB device enable and internal pull-up resistor control bit: 1: Enable USB device transmission and enable internal pull-up resistor; 0: Disable.	0
3	bUC_INT_BUSY	RW	Automatic pause enable bit before USB transmission completion interrupt flag is not cleared: 1: It will automatically pause before interrupt flag UIF_TRANSFER is not cleared. It will automatically respond to the busy NAK in device mode, and will automatically pause the subsequent transmission in host mode; 0: Not pause.	0
2	bUC_RESET_SIE	RW	Software reset control bit of USB protocol processor: 1: Forcibly reset USB protocol processor (SIE); it needs software to clear; 0: Not reset.	1
1	bUC_CLR_ALL	RW	1: Empty USB interrupt flag and FIFO, it needs software to clear; 0: Not empty.	1
0	bUC_DMA_EN	RW	DMA and DMA interrupt control bit of USB: 1: Enable DMA function and DMA interrupt; 0: Disable DMA.	0

USB interrupt enable register (USB_INT_EN)

Bit	Name	Access	Description	Reset value
7	bUIE_DEV_SOF	RW	In USB device mode, receiving SOF packet interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
6	bUIE_DEV_NAK	RW	In USB device mode, receiving NAK interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

5	Reserved	RO	Reserved.	0
4	bUIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	bUIE_HST_SOF	RW	In USB host mode, SOF timing interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	bUIE_SUSPEND	RW	USB bus suspension or wake-up event interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	bUIE_TRANSFER	RW	USB transfer completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	bUIE_DETECT	RW	In USB host mode, USB device connection or disconnection event interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	bUIE_BUS_RST	RW	In USB device mode, USB bus reset event interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

USB device address register (USB_DEV_AD)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0
[6:0]	MASK_USB_ADDR	RW	In host mode, it is the address of the USB device being operated or HUB address; In device mode, it is the address of the USB device.	00h

USB frame number register (USB_FRAME_NO)

Bit	Name	Access	Description	Reset value
[15:0]	USB_FRAME_NO	RO	Frame number, indicates the frame number of the SOF packet to be sent in host mode, and indicates the frame number of the SOF packet currently received in device mode. Among them, the lower 11 bits are the valid frame number, and the higher 3 bits are the micro frame number of high-speed mode.	0

USB_FRAME_NO is a 16-bit register; among them, the lower 11 bits represent the SOF packet frame number, and the higher 3 bits represents the current micro-frame that it belongs to, which can be used for interrupt, synchronous/isochronous transmission under high-speed HUB operation.

USB Suspend Register (USB_SUSPEND)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved.	0
1	bUS_RESUME	RW	Remote wake-up control bit:	0

			1: Wake up the host remotely; 0: No action.	
0	Reserved	RO	Reserved.	0

Note: When remote wake-up is required, pull the bUS_RESUME bit up and then down.

USB speed type register (USB_SPEED_TYPE)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved.	0
[1:0]	USB_SPEED_TYPE	RO	In host mode, it indicates the speed type of the currently connected device. In device mode, it indicates the speed type of the current device; 00: Full speed; 01: High speed; 10: Low speed.	00b

Note: Different from UC_SPEED_TYPE in USB_CTRL register, UC_SPEED_TYPE represents the highest speed expected. Assuming that in device mode, set UC_SPEED_TYPE to high speed. When the device is connected to a host at full speed, the actual speed type is full speed and it can be known by querying USB_SPEED_TYPE register. In host mode, set UC_SPEED_TYPE to high speed. When a device at full speed is connected, the actual communication speed is full speed and it can be known by querying the USB_SPEED_TYPE register.

USB Miscellaneous Status Register (USB_MIS_ST)

Bit	Name	Access	Description	Reset value
7	bUMS_SOF_PRES	RO	SOF packet indication status bit in USB host mode: 1: SOF packet will be sent, and it will be automatically delayed if there are other USB data packets; 0: No SOF packet is sent.	x
6	bUMS_SOF_ACT	RO	SOF packet transmission status bit in USB host mode: 1: SOF packet is being sent out; 0: The transmission is completed or idle.	x
5	bUMS_SIE_FREE	RO	Idle status bit of USB protocol processor: 1: Idle protocol processor; 0: Busy, USB transmission is in progress.	1
4	bUMS_R_FIFO_RDY	RO	USB receiver FIFO data ready status bit: 1: Receiver FIFO is non-empty; 0: Receiver FIFO is empty.	0
3	bUMS_BUS_RESET	RO	USB bus reset status bit: 1: USB bus is at reset status currently; 0: USB bus is at non-reset status currently.	0
2	bUMS_SUSPEND	RO	USB suspension status bit: 1: USB bus is at suspended state, and there is no USB activity for a period of time; 0: USB bus is at non-suspended status.	0
1	bUMS_ATTACH	RO	USB device connection status bit of the port	0

			in USB host mode: 1: The port has been connected to a USB device; 0: No USB device is connected to the port.	
0	bUMS_SPLIT_CAN	RO	In USB host mode, SPLIT packet transmission permission bit: 1: Allow to send SPLIT packet; 1: Disable sending.	0

USB interrupt flag register (USB_INT_FG)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	00b
4	UIF_FIFO_OV	RW1	USB FIFO overflow interrupt flag bit, write 1 to clear it: 1: FIFO overflow trigger; 2. No event.	0
3	UIF_HST_SOF	RW1	SOF timing interrupt flag bit in USB host mode, cleared by writing 1: 1: SOF transmission completion trigger; 2. No event.	0
2	UIF_SUSPEND	RW1	USB bus suspension or wake-up event interrupt flag bit, write 1 to clear it: 1: USB suspension event or wake-up event trigger; 2. No event.	0
1	UIF_TRANSFER	RW1	USB transfer completion interrupt flag bit, write 1 to clear it: 1: A USB transmission completion trigger; 2. No event.	0
0	UIF_DETECT	RW1	In USB host mode, USB device connection or disconnection event interrupt flag bit, write 1 to clear it: 1: USB device connection or disconnection trigger is detected; 2. No event.	0
0	UIF_BUS_RST	RW1	USB bus reset event interrupt flag bit in USB device mode, write 1 to clear it: 1: USB bus reset event trigger; 2. No event.	0

USB interrupt status register (USB_INT_ST)

Bit	Name	Access	Description	Reset value
7	bUIS_IS_NAK	RO	In USB device mode, NAK response status bit: 1: Respond to NAK during current USB transmission; 0: No NAK response.	0
6	bUIS_TOG_OK	RO	Current USB transmission DATA0/1	0

			synchronization flag match status bit: 1: Synchronous; 0: Asynchronous.	
[5:4]	MASK_UIS_TOKEN	RO	In device mode, the token PID of the current USB transfer transaction.	xxb
[3:0]	MASK_UIS_ENDP	RO	In device mode, the endpoint number of the current USB transfer transaction.	xxxxb
[3:0]	MASK_UIS_H_RES	RO	In host mode, the response PID identification of current USB transmission transaction. 0000: Device has no response or timeout; Other values: Response PID.	xxxxb

MASK_UIS_TOKEN is used to identify the token PID of the current USB transmission transaction in USB device mode: 00 means OUT packet; 01 means SOF packet; 10 means IN packet; 11 means SETUP packet.

MASK_UIS_H_RES is only valid in host mode. In host mode, if the host sends OUT/SETUP token packet, the PID is the handshake packet ACK/NAK/STALL/NYET, or the device has no response/timeout. If the host sends IN token packet, the PID is the PID of the data packet (DATA0/DATA1/DATA2/MDATA) or the handshake packet PID.

USB receiving length register (USB_RX_LEN)

Bit	Name	Access	Description	Reset value
[15:0]	USB_RX_LEN	RO	Current count of data received by the USB endpoint, the lower 11 bits are valid, and the higher 5 bits are fixed to 0.	xxxxh

11.2.2 Device Register Description

In USB device mode, CH568 is equipped with 5 groups of bidirectional endpoints 0, 1, 2, 3, 4. The maximum data packet length of all endpoints except endpoint 0 is 512 bytes, and the maximum data packet length of endpoint 0 is 64 bytes.

Endpoint 0 is the default endpoint and supports control transmission. The sending and receiving share a 64-byte data buffer area.

Endpoint 1, endpoint 2, endpoint 3 each includes a sending endpoint IN and a receiving endpoint OUT. The sending and receiving endpoint each has a separate 512-byte data buffer or double 512-byte data buffer respectively, support bulk transmission, interrupt transmission, and isochronous/synchronous transmission.

Endpoint 4 includes a sending endpoint IN and a receiving endpoint OUT. The sending and receiving endpoint each has a separate 512 bytes data buffer respectively, support bulk transmission, interrupt transmission, and isochronous/synchronous transmission.

Endpoints 0/1/2/3 all can set UEPn_DMA register to configure their DMA addresses respectively. Set the endpoint receiving and sending data buffer mode through UEP4_1_MOD and UEP2_3_MOD registers.

Each group of endpoints are equipped with receiving and transmission control register UEPn_TX_CTRL and UEPn_RX_CTRL and sending length register UEPn_T_LEN (n=0/1/2/3/4), which are used to set the synchronization trigger bit of this endpoint, the response to OUT transactions and IN transactions and the length of data to be sent.

As the necessary USB bus pull-up resistor of USB device, it can be set whether to be enabled by software at any time. When bUC_DEV_EN in USB control register USB_CTRL is set to 1, CH568 will set according to

the speed of bUC_SPEED_TYPE, internally connect the pull-up resistor with the DP/DM pin of the USB bus and enable the USB device function.

When a USB bus reset, USB bus suspending or waking event is detected, or when the USB successfully processes data sending or receiving, the USB protocol processor will set corresponding interrupt flag. If the interrupt enable is on, the corresponding interrupt request will be also generated. The application program can directly query or query and analyze the interrupt flag register USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to UIF_BUS_RST and UIF_SUSPEND. In addition, if UIF_TRANSFER is valid, it is required to continue to analyze the USB interrupt status register USB_INT_ST, and perform the corresponding processing according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID identification MASK_UIS_TOKEN. If the synchronization trigger bit bUEP_R_TOG of OUT transaction of each endpoint is set in advance, you can judge whether the synchronization trigger bit of the data packet received matches the synchronization trigger bit of the endpoint through bUIS_TOG_OK. If the data is synchronized, the data is valid. If the data is not synchronized, the data should be discarded. After the USB sending or receiving interrupt is processed each time, the synchronization trigger bit of corresponding endpoint should be modified correctly to detect whether the data packet sent next time and the data packet received next time are synchronized. In addition, bUEP_AUTO_TOG can be set to automatically flip the corresponding synchronization trigger bit after sending or receiving successfully.

The data to be sent by each endpoint is in their own buffer, and the length of the data to be sent is independently set in UEPn_T_LEN, and the sending length of one packet cannot exceed 512 bytes. The data received by each endpoint is in their own buffer, but the length of the data received is in the USB length receiving register USB_RX_LEN, and it can be distinguished according to the current endpoint number when the USB is receiving an interrupt. The maximum packet length that can be received by each endpoint needs to be written into the UEPn_MAX_LEN register in advance.

Table 11-2 List of USB Device Registers

Name	Offset address	Description	Reset value
UEP4_1_MOD	10h	Endpoint 1, 4 mode control register	8h00
UEP2_3_MOD	11h	Endpoint 2, 3 mode control register	8h00
UEP0_DMA	14h	Start address of endpoint 0 buffer	16hxxxx
UEP1_DMA	18h	Start address of endpoint 1 buffer	16hxxxx
UEP2_DMA	1Ch	Start address of endpoint 2 buffer	16hxxxx
UEP3_DMA	20h	Start address of endpoint 3 buffer	16hxxxx
UEP0_MAX_LEN	24h	Endpoint 0 maximum length packet register	16hxxxx
UEP1_MAX_LEN	28h	Endpoint 1 maximum length packet register	16hxxxx
UEP2_MAX_LEN	2Ch	Endpoint 2 maximum length packet register	16hxxxx
UEP3_MAX_LEN	30h	Endpoint 3 maximum length packet register	16hxxxx
UEP4_MAX_LEN	34h	Endpoint 4 maximum length packet register	16hxxxx
UEP0_T_LEN	38h	Endpoint 0 sending length register	16hxxxx
UEP0_TX_CTRL	3Ah	Endpoint 0 sending control register	8h00
UEP0_RX_CTRL	3Bh	Endpoint 0 receiving control register	8h00
UEP1_T_LEN	3Ch	Endpoint 1 sending length register	16hxxxx
UEP1_TX_CTRL	3Eh	Endpoint 1 sending control register	8h00
UEP1_RX_CTRL	3Fh	Endpoint 1 receiving control register	8h00
UEP2_T_LEN	40h	Endpoint 2 sending length register	16hxxxx
UEP2_TX_CTRL	42h	Endpoint 2 sending control register	8h00
UEP2_RX_CTRL	43h	Endpoint 2 receiving control register	8h00

UEP3_T_LEN	44h	Endpoint 3 sending length register	16hxxxx
UEP3_TX_CTRL	46h	Endpoint 3 sending control register	8h00
UEP3_RX_CTRL	47h	Endpoint 3 receiving control register	8h00
UEP4_T_LEN	48h	Endpoint 4 sending length register	16hxxxx
UEP4_TX_CTRL	4Ah	Endpoint 4 sending control register	8h00
UEP4_RX_CTRL	4Bh	Endpoint 4 receiving control register	8h00

USB endpoint 1, 4 mode control register (UEP4_1_MOD)

Bit	Name	Access	Description	Reset value
7	bUEP1_RX_EN	RW	1: Enable endpoint 1 receiving (OUT); 0: Disable endpoint 1 receiving.	0
6	bUEP1_TX_EN	RW	1: Enable endpoint 1 sending (IN); 0: Disable endpoint 1 sending.	0
5	Reserved	RO	Reserved.	0
4	bUEP1_BUF_MOD	RW	Endpoint 1 data buffer mode control bit.	0
3	bUEP4_RX_EN	RW	1: Enable endpoint 4 receiving (OUT); 0: Disable endpoint 4 receiving.	0
2	bUEP4_TX_EN	RW	1: Enable endpoint 4 sending (IN); 0: Disable endpoint 4 sending.	0
[1:0]	Reserved	RO	Reserved.	0

The data buffer modes of USB endpoints 0 and 4 are configured by a combination of bUEP4_RX_EN and bUEP4_TX_EN. Refer to the following table for details:

Table 11-3 Endpoint 0 and 4 Buffer Mode

bUEP4_RX_EN	bUEP4_TX_EN	Description: arrange from low to high with UEP0 DMA as start address
0	0	Endpoint 0 single 64-byte transceiving shared buffers (IN and OUT), and endpoint 4 is disabled for transceiving.
1	0	Endpoint 0 single 64-byte transceiving shared buffers; endpoint 4 single 512-byte receiving buffer (OUT).
0	1	Endpoint 0 single 64-byte transceiving shared buffers; endpoint 4 single 512-byte sending buffer (IN).
1	1	Endpoint 0 single 64-byte transceiving shared buffers; endpoint 4 single 512-byte receiving buffer (OUT); Endpoint 4 single 512-byte receiving buffer (IN). All 1088 bytes are arranged as follows: UEP0_DMA+0 address: 64-byte start address of endpoint 0 transceiving shared buffer; UEP0_DMA+64 address: 512-byte start address of endpoint 4 receiving buffer; UEP1_DMA+64+512 address: 512-byte start address of endpoint 4 sending buffer.

USB endpoint 2, 3 mode control register (UEP2_3_MOD)

Bit	Name	Access	Description	Reset value
7	bUEP3_RX_EN	RW	1: Enable endpoint 3 receiving (OUT); 0: Disable endpoint 3 receiving.	0
6	bUEP3_TX_EN	RW	1: Enable endpoint 3 sending (IN); 0: Disable endpoint 3 sending.	0

5	Reserved	RO	Reserved.	0
4	bUEP3_BUF_MOD	RW	Endpoint 3 data buffer mode control bit.	0
3	bUEP2_RX_EN	RW	1: Enable endpoint 2 receiving (OUT); 0: Disable endpoint 2 receiving.	0
2	bUEP2_TX_EN	RW	1: Enable endpoint 2 sending (IN); 0: Disable endpoint 2 sending.	0
1	Reserved	RO	Reserved.	0
0	bUEP2_BUF_MOD	RW	Endpoint 2 data buffer mode control bit.	0

The data buffer modes of USB endpoints 1, 2 and 3 are controlled by a combination of bUEPn_RX_EN, bUEPn_TX_EN and bUEPn_BUF_MOD (n=1/2/3) respectively, refer to the following table for details. Among them, in double 512-byte buffer mode, the first 512-byte buffer will be selected based on bUEP_*_TOG=0 and the last 512-byte buffer will be selected based on bUEP_*_TOG=1 during USB data transmission, and bUEP_AUTO_TOG=1 is set to realize automatic switch.

Table 11-4 Endpoint n Buffer Mode (n=1/2/3)

bUEPn_RX_EN	bUEPn_TX_EN	bUEPn_BUF_MOD	Description: arrange from low to high with UEPn_DMA as start address
0	0	x	Endpoint is disabled, and UEPn_DMA buffer is not used.
1	0	0	Single 512-byte receiving buffer (OUT).
1	0	1	Double 512-byte receiving buffer (OUT), selected by bUEP_R_TOG.
0	1	0	Single 512-byte sending buffer (IN).
0	1	1	Double 512-byte sending buffer (IN), selected by bUEP_T_TOG.
1	1	0	Single 512-byte receiving buffer (OUT), single 512-byte sending buffer (IN).
1	1	1	Double 512-byte receiving buffer (OUT), selected by bUEP_R_TOG. Double 512-byte sending buffer (IN), selected by bUEP_T_TOG. All 2K bytes are arranged as follows: UEPn_DMA+0 address: Endpoint receiving address when bUEP_R_TOG=0; UEPn_DMA+512 address: Endpoint receiving address when bUEP_R_TOG=1; UEPn_DMA+1024 address: Endpoint sending address when bUEP_T_TOG=0; UEPn_DMA+1536 address: Endpoint sending address when bUEP_T_TOG=1.

Start address of USB endpoint n buffer (UEPn_DMA)(n=1/2/3):

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_DMA	RW	Start address of endpoint n buffer, the lowest 2 bits are fixed to 0 (4 bytes are aligned), and the higher 1 bit is fixed to 0.	xxxxh

Endpoint n maximum length packet register (UEPn_MAX_LEN) (n=1/2/3)

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_MAX_LEN	RW	Maximum packet length of data received by endpoint n.	xxxxh

Note: This maximum packet length determines the maximum length of data that can be received by the endpoint. The data beyond this length will be discarded and DMA will not be sent to the custom area.

Endpoint n sending length register (UEPn_T_LEN):

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_T_LEN	RW	Set the number of bytes of data to be sent by USB endpoint n, the lower 10 bits are valid, and the higher 6 bits are fixed to 0, with a maximum length of 512.	xxxxh

Endpoint n sending control register (UEPn_TX_CTRL):

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	bUEP_AUTO_TOG	RW	Synchronization trigger bit auto flip enable control bit: 1: After data is successfully sent, the corresponding synchronization trigger bit is automatically flipped; 0: It is not flipped automatically, but can be switched manually. Only endpoint 1/2/3 supports, and isochronous/synchronous transmission can only be switched manually.	0
[4:3]	MASK_UEP_T_TOG	RW	Synchronization trigger bit of the transmitter (processing IN transactions) of USB endpoint n 00: Send DATA0; 01: Send DATA1; 10: Send DATA2; 11: Send MDATA;	0
2	bUEP_T_RES_NO	RW	1: Expect no response, used to achieve isochronous/synchronous transmission of endpoints other than endpoint 0. Ignore MASK_UEP_T_RES at this time; 0: Expect response.	0
[1:0]	MASK_UEP_T_RES	RW	Response control from transmitter of endpoint n to IN services: 00: Data is ready and ACK is expected; 10: Response NAK or busy; 11: Response STALL or error.	0

Endpoint n receiving control register (UEPn_RX_CTRL):

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	bUEP_AUTO_TOG	RW	Synchronization trigger bit auto flip enable	0

			control bit: 1: After data is successfully received, the corresponding synchronization trigger bit is automatically flipped; 0: It is not flipped automatically, but can be switched manually. Only endpoint 1/2/3 supports, and isochronous/synchronous transmission can only be switched manually.	
[4:3]	MASK_UEP_R_TOG	RW	Expected synchronization trigger bit of the receiver (processing OUT transactions) of USB endpoint n: 00: Expect DATA0; 01: Expect DATA1; 10: Expect DATA2; 11: Expect MDATA. It is invalid for isochronous/synchronous transmission.	0
2	bUEP_R_RES_NO	RW	1: Expect no response, used to achieve isochronous/synchronous transmission of endpoints other than endpoint 0. Ignore MASK_UEP_R_RES at this time; 0: Expect response.	0
[1:0]	MASK_UEP_R_RES	RW	Response control from receiver of USB endpoint n to OUT transactions: 00: Response ACK; - 10: Response NAK or busy; 11: Response STALL or error; 01: Response NYET. It is invalid for isochronous/synchronous transmission.	0

11.2.3 USB Host Register

In USB host mode, CH568 is equipped with 1 set of bidirectional host endpoints, including a sending endpoint OUT and a receiving endpoint IN. The maximum data packet length is 512 bytes, support control transmission, interrupt transmission, bulk transmission and isochronous/synchronous transmission.

Each USB transaction initiated by host endpoint always automatically sets the interrupt flag UIF_TRANSFER after processing. The application program can directly query or query and analyze the interrupt flag register USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to each interrupt flag. In addition, if UIF_TRANSFER is valid, it is required to continue to analyze the USB interrupt status register USB_INT_ST, and perform the corresponding processing according to the response PID identification MASK_UIS_H_RES of the current USB transmission transaction.

If the synchronization trigger bit bUH_R_TOG of IN transaction of host receiving endpoint is set in advance, you can judge whether the synchronization trigger bit of the data packet received matches the synchronization trigger bit of the endpoint through bUIS_TOG_OK. If the data is synchronized, the data is valid. If the data is not synchronized, the data should be discarded. After the USB sending or receiving interrupt is processed each time, the synchronization trigger bit of corresponding host endpoint should be

modified correctly to synchronize the data packet sent next time and detect whether the data packet received next time is synchronized. In addition, bUEP_AUTO_TOG can be set to automatically flip the corresponding synchronization trigger bit after sending or receiving successfully.

USB host token setting register UH_EP_PID is used to set the endpoint number of the target device being operated and the token PID packet identification of the USB transmission transaction. The data corresponding to the SETUP token and OUT token is provided by the host sending endpoint. The data to be sent is in the UH_TX_DMA buffer, and the length of the data to be sent is set in UH_TX_LEN. The data corresponding to the IN token is returned by the target device to the host receiving endpoint, the received data is stored in the UH_RX_DMA buffer, and the received data length is stored in USB_RX_LEN. The maximum packet length that can be received by the host endpoint needs to be written to the UH_RX_MAX_LEN register in advance.

Table 11-5 List of USB Host Related Registers

Name	Offset address	Description	Reset value
UHOST_CTRL	01h	USB host control register	8h00
UH_EP_MOD	11h	USB host endpoint mode control register	8h00
UH_RX_DMA	1Ch	USB host receiving buffer area start address	16hxxxx
UH_TX_DMA	20h	USB host sending buffer area start address	16hxxxx
UH_RX_MAX_LEN	2Ch	USB host receiving maximum length packet register	16hxxxx
UH_SETUP	3Eh	USB host auxiliary setting register	8h00
UH_EP_PID	40h	USB host token setting register	8h00
UH_RX_CTRL	43h	USB host receiving endpoint control register	8h00
UH_TX_LEN	44h	USB host sending length register	16hxxxx
UH_TX_CTRL	46h	USB host sending endpoint control register	8h00
UH_SPLIT_DATA	48h	USB host sending SPLIT packet data	16hxxxx

USB host control register (UHOST_CTRL)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved.	0
2	bUH_TX_BUS_RESUME	RW	In host mode, it indicates that host wakes up the device.	0
1	bUH_TX_BUS_SUSPEND	RW	USB host sending suspend signal.	0
0	bUH_TX_BUS_RESET	RW	The USB host sending bus reset signal.	0

Note: The reset time is determined by the high level duration of bUH_TX_BUS_RESET. If the host wakes up the device, it is determined by the bUH_TX_BUS_RESUME edge method, so it is only required to pull bUH_TX_BUS_RESUME high and then low for wake-up.

USB host endpoint mode control register (UH_EP_MOD)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	bUH_EP_TX_EN	RW	Host sending endpoint send (SETUP/OUT) enable bit: 1: Enable endpoint sending; 0: Disable endpoint sending.	0
5	Reserved	RO	Reserved.	0
4	bUH_EP_TBUF_MOD	RW	Host sending endpoint send data buffer	0

			mode control bit.	
3	bUH_EP_RX_EN	RW	Host receiving endpoint receive (IN) enable bit: 1: Enable endpoint receiving; 0: Disable endpoint receiving.	0
[2:1]	Reserved	RO	Reserved.	00b
0	bUH_EP_RBUF_MOD	RW	USB host receiving endpoint receive data buffer area mode control bit.	0

The data buffer modes of USB host sending endpoint are controlled by a combination of bUH_EP_TX_EN and bUH_EP_TBUF_MOD, refer to the following table.

Table 11-6 Host send buffer Mode

bUH_EP_TX_EN	bUH_EP_TBUF_MOD	Structure description: Take UH_TX_DMA as start address
0	x	Endpoint is disabled, UH_TX_DMA buffer is not used.
1	0	Single 512-byte sending buffer (SETUP/OUT).
1	1	Double 512-byte sending buffer, selected by bUH_T_TOG: When bUH_T_TOG=0, select the first 512 bytes of buffer; When bUH_T_TOG=1, select the last 512 bytes of buffer.

The data buffer modes of USB host receiving endpoint are controlled by a combination of bUH_EP_RX_EN and bUH_EP_RBUF_MOD, refer to the following table.

Table 11-7 Host receive buffer Mode

bUH_EP_RX_EN	bUH_EP_RBUF_MOD	Structure description: Take UH_TX_DMA as start address
0	x	Endpoint is disabled, UH_RX_DMA buffer is not used.
1	0	Single 512-byte receiving buffer (IN).
1	1	Double 512-byte receiving buffer, selected by bUH_R_TOG: When bUH_R_TOG=0, select the first 512 bytes of buffer; When bUH_R_TOG=1, select the last 512 bytes of buffer.

Start address of USB host receiving buffer (UH_RX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	UH_RX_DMA	RW	Start address of host endpoint data receiving buffer, the lowest 2 bits are fixed to 0 (4 bytes are aligned), and the highest 1 bit is fixed to 0.	xxxxh

Start address of USB host sending buffer (UH_TX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	UH_TX_DMA	RW	Start address of host endpoint data sending buffer, the lowest 2 bits are fixed to 0 (4 bytes are aligned), and the highest 1 bit is fixed to 0.	xxxxh

USB host receiving maximum length packet register (UH_RX_MAX_LEN)

Bit	Name	Access	Description	Reset value
[15:0]	UH_RX_MAX_LEN	RW	Maximum packet length of data received by	xxxxh

			the host endpoint.	

Note: This maximum packet size determines the maximum length of data that can be received by the endpoint. The data beyond this length will be discarded and DMA will send it to the custom area.

USB host auxiliary setting register (UH_SETUP):

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0
6	bUH_SOF_EN	WO	Automatically generate SOF packet enable control bit: 1: The host automatically generates SOF packet; 0: No SOF packet is generated.	0
[5:0]	Reserved	RO	Reserved.	00

USB host token setting register (UH_EP_PID)

Bit	Name	Access	Description	Reset value
[7:4]	MASK_UH_TOKEN	RW	Set the token PID packet identification of this USB transmission transaction.	0000b
[3:0]	MASK_UH_ENDP	RW	Set the endpoint number of the target device being operated this time.	0000b

USB host receiving endpoint control register (UH_RX_CTRL)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0b
6	bUH_R_DATA_NO	RW	1: Data packet not expected, used for high-speed HUB operation in host mode; 0: Data packet expected (IN).	0
5	bUH_R_AUTO_TOG	RW	Synchronization trigger bit auto flip enable control bit: 1: After the data is successfully received, the corresponding expected synchronization trigger bit is automatically flipped; 0: It is not flipped automatically, but can be switched manually.	0
[4:3]	MASK_UH_R_TOG	RW	Synchronization trigger bit expected by the host receiver (processing IN transactions), 00: Expect DATA0; 01: Expect DATA1; 10: Expect DATA2; 11: Expect MDATA.	0
2	bUH_R_RES_NO	RW	1: No response, used to achieve isochronous/synchronous transmission of endpoints other than endpoint 0. Ignore MASK_UEP_R_RES at this time; 0: Send response after data is received successfully.	0
[1:0]	MASK_UH_R_RES	RW	Response control bit from receiver of host to IN transactions:	00b

			00: Response ACK; - It is invalid for isochronous/synchronous transmission.	

USB host sending length register (UH_TX_LEN)

Bit	Name	Access	Description	Reset value
[15:0]	UH_TX_LEN_H	RW	Set the number of bytes of data to be sent by USB host sending endpoint, only the lower 11 bits are valid, and the higher 5 bits are fixed to 0.	xxxxh

USB host sending endpoint control register (UH_TX_CTRL)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0b
6	bUH_T_DATA_NO	RW	1: Not send data packet (PING/SPLIT); 0: Send data packet (OUT/SETUP).	0
5	bUH_T_AUTO_TOG	RW	Synchronization trigger bit auto flip enable control bit: 1: After the data is successfully received, the corresponding synchronization trigger bit is automatically flipped; 0: It is not flipped automatically, but can be switched manually.	0
[4:3]	MASK_UH_T_TOG	RW	Synchronization trigger bit prepared by USB host transmitter (processing SETUP/OUT transactions) 00: Send DATA0; 01: Send DATA1; 10: Send DATA2; 11: Send MDATA.	00b
2	bUH_T_RES_NO	RW	1: No response, used to achieve isochronous/synchronous transmission of endpoints other than endpoint 0. Ignore MASK_UEP_T_RES at this time; 0: Expect response after sending data successfully.	0
[1:0]	MASK_UH_T_RES	RW	Response control bit from USB host transmitter to SETUP/OUT transaction 00: Expect response ACK; 10: Expect response NAK or busy; 11: Expect response STALL or error; 01: Expect response to NYET. It is invalid for isochronous/synchronous transmission.	00b

USB host sending SPLIT packet data (UH_SPLIT_DATA)

Bit	Name	Access	Description	Reset value
[15:0]	UH_SPLIT_DATA	RW	Data content of SPLIT packet sent by the	0xxxh

			host endpoint, the lower 12 bits are valid, and the higher 4 bits are fixed to 0	

11.3 USB Device Mode Configuration

11.3.1 Basic Initial Configuration

1. Set USB_CTRL register, set bUC_HOST_MODE bit to 0, and configure USB device mode;
2. Set USB_CTRL register, clear bUC_RESET_SIE and bUC_CLR_ALL to 0, set bUC_INT_BUSY and bUC_DMA_EN to 1, configure UC_SPEED_TYPE to select the speed of USB device. If it is set as high-speed device, but the current host is at full speed, the controller will automatically slow down and switch to full speed, and the actual communication speed can be queried in the USB_SPEED_TYPE register.
3. Clear the device address register USB_DEV_AD and interrupt flag register USB_INT_FG, optional operation, enable the required interrupt, and write into USB_INT_EN register;
4. Configure the device endpoint data transceiving buffer mode register UEP4_1_MOD/ UEP2_3_MOD, and the transceiving control register UEPn_TX_CTRL/ UEPn_RX_CTRL;
5. Set the endpoint maximum packet receiving length UEPn_MAX_LEN register and the endpoint data transceiving start address UEPn_DMA;
6. Set bUC_DEV_PU_EN bit of USB_CTRL register to 1, and enable USB device function.

11.4 USB Host Mode Configuration

11.4.1 Basic Initial Configuration

1. Set USB_CTRL register, set bUC_HOST_MODE bit to 1, and configure USB device mode;
2. Set USB_CTRL register, clear bUC_RESET_SIE and bUC_CLR_ALL to 0, set bUC_INT_BUSY and bUC_DMA_EN to 1, configure UC_SPEED_TYPE to select the speed of USB device. If it is set as high-speed device, but the current connecting device is at full speed, the controller will automatically slow down and switch to full speed, and the actual communication speed can be queried in the USB_SPEED_TYPE register.
3. Clear the device address register USB_DEV_AD and interrupt flag register USB_INT_FG, optional operation, enable the required interrupt, and write into USB_INT_EN register;
4. Configure the host endpoint data transceiving buffer mode register UH_EP_MOD, and the transceiving control register UH_RX_CTRL / UH_TX_CTRL;
5. Set the host endpoint maximum packet receiving length UH_RX_MAX_LEN register and the host endpoint data transceiving start address UH_RX_DMA / UH_TX_DMA;
6. Set the bUH_SOF_EN bit of UH_SETUP register to 1, and enable the port to automatically transmit SOF packets.

Chapter 12 SD Controller and AES/SM4 Module

12.1 Introduction to SD Controller and AES/SM4 Module

CH568 chip is equipped with 4 independent SD controllers: SD0, SD1, SD2 and SD3. Compared with common controllers, it is provided with additional encryption/decryption algorithm module support, which can meet the data security requirements of the market.

The main features are as follows:

1) Support SD physical layer 1.0, 2.0 specifications, support UHS-I SDR50 mode (forward compatible) of SD3.0 specifications;
2) Conform to 4.4 and 4.5.1 specifications of eMMC card, and compatible with 5.0 specifications and HS200 mode;
3) 4 controllers all support the single-wire and four-wire mode of eMMC card, and SD0 and SD2 support the single-wire, four-wire and eight-wire mode of eMMC card;
4) Support SD card, SDIO card, eMMC card and other devices that comply with SD protocol;
5) Support SD interface data for AES and SM4 algorithm encryption and decryption;
6) The 4 controllers work independently, support DMA and interrupts.

12.2 SD Register Description

CH568 is equipped with 4 independent SD controllers, and each controller has a similar control unit.
SD0 related register physical base address: 0x0040 A000
SD1 related register physical base address: 0x0040 A040
SD2 related register physical base address: 0x0040 A080
SD3 related register physical base address: 0x0040 A0C0

Table 12-1 List of SD Registers

Name	Offset address	Description	Reset value
SD_CLK_CFG	3Ch	Clock configuration register	16h0214
SDx_ARGUMENT	00h	Command parameter register	32h00000000
SDx_CMD_SET	04h	Command setting register	16h0000
SDx_RESPONSE0	08h	Response parameter register 0	32h00000000
SDx_RESPONSE1	0Ch	Response parameter register 1	32h00000000
SDx_RESPONSE2	10h	Response parameter register 2	32h00000000
SDx_RESPONSE3	14h	Response parameter register 3	32h00000000
SDx_WRITE_CONT	14h	Continue write start register	32h00000000
SDx_CTRL	18h	Control register	8h15
SDx_TOCNT	1Ch	Timeout counter register	8h0C
SDx_STATUS	20h	Status register	32h00000000
SDx_INT_FG	24h	Interrupt flag register	16h0000
SDx_INT_EN	28h	Interrupt enable register	16h0000
SDx_DMA	2Ch	DMA start address register	16hxxxx
SDx_BLOCK_CFG	30h	Transmission block configuration register	32h00000000
SDx_TRAN_MODE	34h,	Transmission mode register	8h00

Clock configuration register (SD_CLK_CFG)

Bit	Name	Access	Description	Reset value

[16:10]	Reserved	RO	Reserved.	00h
9	bSDCLK_Mode	WO	Clock frequency mode selection bit: 1: High-speed mode, 25M-100MHz; 0: Low speed mode, 400KHz.	1
8	bSDCLK_OE	WO	SD physical clock signal line output control bit: 1: On, output communication clock; 0: Off.	0
[7:5]	Reserved	RO	Reserved.	0
[4:0]	MASK_SD_CLK_PRE	WO	SD controller clock (SDCLK) division factor: When bSDCLK_Mode=1, then SDCLK = 480M/MASK_CLK_PRE; When bSDCLK_Mode =0, then SDCLK = 480M/MASK_CLK_PRE/64. Writing 1 is equivalent to turning off the SDC module sampling clock.	14h

Note: The clock configured by clock configuration register (SD_CLK_CFG) is shared by 4 SD card controllers. That is, 4 SD card controller modules work at the same clock frequency.

Command parameter register (SDx_ARGUMENT) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[31:0]	SD_ARGUMENT	RW	SD/eMMC 32-bit command parameter register.	0

Command setting register (SDx_CMD_SET) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved.	0
11	bCHK_RESP_IDX	RW	Command index for check response: 1: Required; 0: Not required.	0
10	bCHK_RESP_CRC	RW	CRC for check response: 1: Required; 0: Not required.	0
[9:8]	MASK_RESP_TYPE	RW	Expected response type: 00b: No response; 01b: The response length is 136 bits; 10b: The response length is 48 bits; 11b: The response length is 48 bits, and it is R1b type response.	0
[7:6]	Reserved	RO	Reserved.	0
[5:0]	MASK_CMD_IDX	RW	The index number of the command being sent currently.	0

Response parameter register (SDx_RESPONSE) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[31:0]	SDx_RESPONSE0	RO	Response parameter register 0	0
[63:32]	SDx_RESPONSE1	RO	Response parameter register 1	0
[95:64]	SDx_RESPONSE2	RO	Response parameter register 2	0

[127:96]	SDx_RESPONSE3	RO	Response parameter register 3	0
[127:96]	SDx_WRITE_CONT	WO	Multiplexing SD_RESPONSE3 register, used to start write operation in the multi-block writing process.	0

Note: When the response length is 136 bits, the effective data is 128 bits. When the response length is 48 bits, the effective data length is 32 bits. SDx_RESPONSEx register is used to store the effective data parameters for response.

SDx_RESPONSE3 register is multiplexed. The multiplexing of the register is: in the process of continuously writing multiple blocks of data to the card using the CMD25 command and when the block interrupt is completed, if it is not required to change DMA address, write the action of this register and start the operation of writing data to SD. To change DMA address, write to the DMA address register to start to write data to SD, and do not need to start it by writing to register.

Control register (SDx_CTRL) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved.	0
5	bSC_NEG_SAMPLE	RW	Cmd and Data signal line sampling mode selection bits: 1: Sampling on the falling edge; 1: Sampling on the rising edge.	0
4	bSC_RST_DAT_LGC	RW	1: Reset the internal data transceiving logic, needs software to clear; 0: Work normally;	1
3	bSC_DMA_ENABLE	RW	DMA and DMA interrupt control bit of SD controller: 1: Enable DMA function and DMA interrupt; 0: Switch off DMA.	0
2	bSC_ALL_CLR	RW	1: Reset SD controller logic, needs software to clear; 0: Work normally;	1
[1:0]	DAT_LINE_WIDTH	RW	Data line width of the logic sampling for transceiving data (communication data line width): 00: The receiver-transmitter only uses dat[0], single data line; 01: The receiver-transmitter uses dat[3:0], 4 data lines; 10: The receiver-transmitter uses dat[7:0], 8 data lines. This value is only supported by 0# 2# controller, used for the 8-wire mode of eMMC card.	01b

Timeout control register (SDx_TOCNT) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved.	0
[3:0]	MASK_TOCNT	RW	Response/data timeout configuration: 0: Disable the internal timeout mechanism; Non-zero: Set the timeout time, and valid	Ch

			values are 0-12. Calculation method: SD card clock cycle * 4194304 * MASK_TOCNT. For example: If SDCLK cycle is 10ns at this time, write 12, and the timeout time is 10ns * (4194304) * (12) = 503ms.	

Note: 1. The above data timeout includes the following 4 situations:

1) DAT[0] busy timeout after R1b response;
2) When writing data block, DAT[0] busy timeout after CRC status;
3) When writing data block, waiting for CRC status timeout;
4) When reading data block, waiting for start bit timeout.

2. The response to the command also supports timeout mechanism. If the response times out, it will be given by SIF_RE_TMOUT interrupt in the interrupt register. The command timeout uses the maximum timeout value given by the protocol: 64 Tsdclk.

Status indication register (SDx_STATUS) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[32:18]	Reserved	RO	Reserved.	0
17	bST_DAT0_HI	RO	1: The current DAT0 line is at high level 0: Low level.	0
16	bST_CMD_HI	RO	1: The current CMD line is at high level; 0: Low level.	0
[15:0]	MASK_BLOCK_NUM	RO	It indicates the number of blocks that have been successfully transmitted in the current multi-block transmission operation.	0

Interrupt flag register (SDx_INT_FG) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved.	0
9	bSIF_SDIO_INT	RW1	SDIO card interrupt flag bit, cleared by writing 1: 1: SDIO card generates card interrupt; 0: No event.	0
8	bSIF_FIFO_OF	RW1	FIFO overflow interrupt flag bit, cleared by writing 1: 1: FIFO overflow trigger; 0: No event.	0
7	bSIF_BLOCK_GAP	RW1	Single block transmission completion flag bit, cleared by writing 1: 1: Single block receiving and transmission completion trigger; 0: No event.	0
6	bSIF_TRANS_SC	RW1	Flag bit for completing transmitting the number of requested blocks, cleared by writing 1: 1: The number of requested blocks is triggered when transmission is completed; 0: No event.	0

5	bSIF_TRANS_ER	RW1	Transmission CRC error flag bit, cleared by writing 1: 1: CRC error trigger; 0: No event.	0
4	bSIF_DATA_TMO	RW1	Data timeout flag bit, cleared by writing 1: 1: Data timeout trigger; 0: No event.	0
3	bSIF_CMD_DONE	RW1	Command completion flag bit, cleared by writing 1: 1: Send the command, and receive the completion of response; 0: No event.	0
2	bSIF_RE_IDX_ER	RW	Response index number check error flag bit, cleared by writing 1: 1: Response index number check error trigger; 0: No event.	0
1	bSIF_RE_CRC_WR	RW	Response CRC check error flag bit, cleared by writing 1: 1: Response CRC check error trigger; 0: No event.	0
0	bSIF_RE_TMOUT	RW	Receiving response timeout flag bit, cleared by writing 1: 1: Response timeout trigger; 0: No event.	0

Interrupt enable register (SDx_INT_EN) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
9	bSIE_SDIO_INT	RW	SDIO card interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
8	bSIE_FIFO_OF	RW	FIFO overflow interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
7	bSIE_BLOCK_GAP	RW	Single block completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
6	bSIE_TRANS_SC	RW	Requested block transmission completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
5	bSIE_TRANS_ER	RW	Block transmission CRC error interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
4	bSIE_DATA_TMO	RW	Data timeout interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	bSIE_CMD_DONE	RW	Command completion interrupt:	0

			1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	
2	bSIE_RE_IDX_ER	RW	Response index check error interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	bSIE_RE_CRC_WR	RW	Response CRC check error interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	bSIE_RE_TMOUT	RW	Command response timeout interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

Data block DMA start address register (SDx_DMA) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:0]	SD_DMA	RW	Start address of reading and writing data buffer, the lower 3 bits are fixed to 0 (8 bytes are aligned).	0000h

Note: When reading data from SD, this register stores the start address of the read data in SRAM. When writing data to SD card, the start address of the data to be written in the SRAM is stored.

If continuous multi-block read and write SD operations are performed, the user can write to SDx_DMA register to change the DMA address as needed after the single block transmission is completed (bSIF_BLOCK_GAP). Do not change the DMA address during the transmission process, otherwise, data counting errors may be caused.

When performing continuous multi-block writing, it is required to start the continued write operation by writing to SDx_WRITE_CONT or SDx_DMA register after the single block transmission is completed. It is not required when reading multi-block.

Transmission block configuration register (SDx_BLOCK_CFG) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved.	0
[27:16]	BLOCK_SIZE	RW	Single block transmission size (1-2048 bytes).	0
[15:0]	BLOCK_NUM	RW	Count of blocks to be transmitted by DMA this time (1~65535 blocks), automatically cleared internally. If the number of blocks is not zero, the receiving or sending is enabled.	0

Transmission mode register (SDx_TRAN_MODE) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved.	0
2	bTM_EMMC_BOOT	RW	Set the eMMC card transmission mode: 1: Boot mode; 0: Normal mode. Note: Only for eMMC card.	0
1	bTM_GAP_STOP	RW	Rising edge (0->1): Enable data block clock counting (the clock will automatically pause after a complete data block is recorded every	0

			time, and needs software to clear and restart the clock); Other: Invalid.	
0	bTM_WR_SD	RW	Direction of DMA transmission: 1: Controller to SD; 0: SD to controller.	0

Note: When bCLK_COUNT is on, the clock will be on automatically when sending command if it is automatically paused, and it will be paused again after the command is completed (including response and command timeout conditions).

12.3 SD Control Application

12.3.1 SD Command Sending Operation:

1. Set the 32-bit SDx_ARGUMENT parameter register;
2. Set the 16-bit SDx_CMD_SET register;
3. Wait for the command sending status, and query SDx_INT_FG register. If the command is sent successfully, it will generate the command transmission success flag. Otherwise, it will generate CRC error, or timeout, or response index error flag.

12.3.2 Operation of Reading SD Card multi-Block Data:

1. Set SDx_DMA register, set the DMA transmission direction of SD0x_TRAN_MODE register from SD to the controller, set SD_BLOCK_CFG (the number of bytes received by per block, and the number of blocks to be received by DMA this time), and the controller is now ready to start receiving the data block returned by SD card.
2. Set the 32-bit SDx_ARGUMENT parameter register and SDx_CMD_SET register, and issue CMD18 (read multi-block command).
3. Wait for the command to be sent.
4. After the controller has successfully received N blocks, transmission success interrupt may be generated (bSIF_TRANS_SC=1). If a transmission error occurs in the process, the corresponding error interrupt will be generated. At this time, read the status register (SDx_STATUS) to understand the number of blocks that have been successfully transmitted this time.

12.3.3 Operation of Writing SD Card multi-Block Data:

1. Set the 32-bit SDx_ARGUMENT parameter register and SDx_CMD_SET register, and issue CMD25 (write multi-block command).
2. Wait for the command to be sent.
3. Set SDx_DMA register, set the DMA transmission direction of SDx_TRAN_MODE register from the controller to SD, set SDx_BLOCK_CFG (the number of bytes sent by per block, and the number of blocks to be sent by DMA this time), and the controller now starts to send the data block to SD card.
4. After the controller has successfully sent N blocks, transmission success interrupt may be generated (bSIF_TRANS_SC=1). If a transmission error occurs in the process, a transmission error interrupt will be generated. At this time, read the status register (SDx_STATUS) to understand the number of blocks that have been successfully transmitted this time.

12.4 AES/SM4 Module Function Description

CH568 is equipped with a built-in block cipher algorithm module, which supports two types of block cipher algorithms (AES and SM4), and electronic codebook (ECB) and counter (CTR) modes. There are totally 8 combinations as follows:

ECB mode and CTR mode with SM4 algorithm 128bit key;

ECB mode and CTR mode with AES algorithm 128bit key;
ECB mode and CTR mode with AES algorithm 192bit key;
ECB mode and CTR mode with AES algorithm 256bit key;

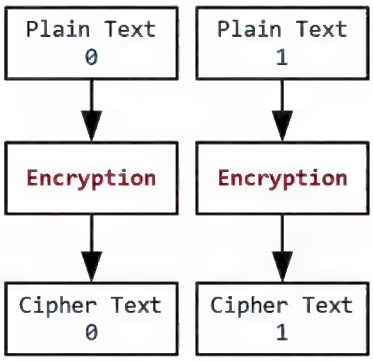
12.4.1 AES/SM4 Algorithm

The AES (Advanced Encryption Standard) algorithm is a block encryption method that uses a symmetric block cipher system, which is one of the most popular algorithms in symmetric key encryption. The SM4 block cipher algorithm is generally a special block cipher algorithm for wireless local area networks and trusted computers, and it can also be used for data encryption protection in other environments.

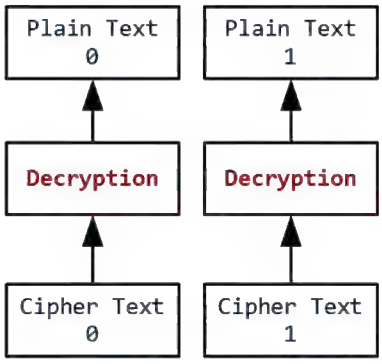
In the process of data encryption and decryption, the key needs to be loaded. For AES algorithm, the key length is set to 128/192/256 bits, the user key is extended to 11×128/13×128/15×128-bit extended keys. While for SM4 algorithm, the 128-bit user key is extended to 32×32-bit extended key. These extended keys are stored in internal registers for use during encryption and decryption.

12.4.2 ECB and CTR Mode

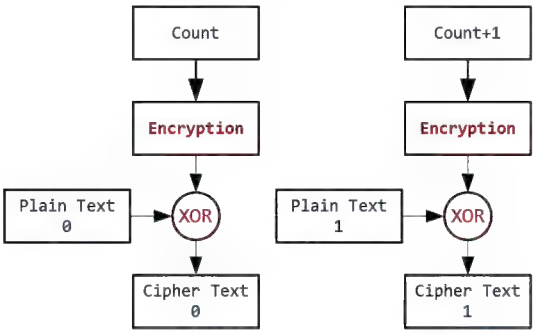
AES/SM4 supports two modes, including electronic codebook (ECB) mode and counter (CTR) mode. Among them, the security performance of CTR mode is higher than that of ECB mode. The difference between the two is shown in Figure 13-1. In ECB mode, there is a one-to-one correspondence between plain text and cipher text, and the encrypted plain text is directly used as cipher text. While in CTR mode, a 128-bit counter value needs to be loaded in advance to encrypt the count value, and the encrypted count value and plain text are as the cipher text through exclusive OR. It is worth noting that in CTR decryption mode, only the count value is encrypted, but not decrypted.



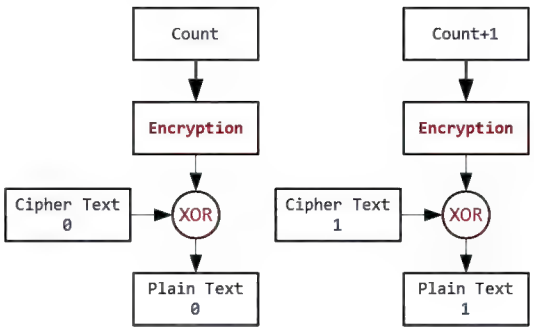
(a) Encryption in ECB mode



(b) Decryption in ECB mode



(c) Encryption in CTR mode



(d) Decryption in CTR mode

Figure 12-1 Encryption and Decryption Diagram in ECB and CTR Modes

12.5 AES/SM4 Module Register Description

AES/SM4 module related register physical base address: 0x0040 c400

Table 12-2 List of AES/SM4 Module Registers

Name	Offset address	Description	Reset value
AES_SM4_CTRL	0x00	AES/SM4 control register	32h20
AES_SM4_INT_FG	0x04	AES/SM4 interrupt flag register	32h0
AES_SM4_KEY7	0x08	Key register 7	32hxxxxxxxx
AES_SM4_KEY6	0x0C	Key register 6	32hxxxxxxxx
AES_SM4_KEY5	0x10	Key register 5	32hxxxxxxxx
AES_SM4_KEY4	0x14	Key register 4	32hxxxxxxxx
AES_SM4_KEY3	0x18	Key register 3	32hxxxxxxxx
AES_SM4_KEY2	0x1C	Key register 2	32hxxxxxxxx
AES_SM4_KEY1	0x20	Key register 1	32hxxxxxxxx
AES_SM4_KEY0	0x24	Key register 0	32hxxxxxxxx
AES_SM4_IV3	0x28	Count value register 3	32hxxxxxxxx
AES_SM4_IV2	0x2C	Count value register 2	32hxxxxxxxx
AES_SM4_IV1	0x30	Count value register 1	32hxxxxxxxx
AES_SM4_IV0	0x34	Count value register 0	32hxxxxxxxx

AES/SM4 control register (AES_SM4_CTRL)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved.	0
16	bKEYE_ACT_IE	RW	Key expansion completion interrupt enable: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
[15:12]	Reserved	RO	Reserved.	0
[11:10]	MASK_Key_LEN	RW	Key length setting: 00:128-bit; 01:192-bit; 10:256-bit; 11: Reserved.	0
9	bBCIPHER_MOD	RW	Block cipher mode selection bit: 1: CTR mode; 0: ECB mode.	0
8	bALGRM_MOD	RW	Algorithm mode selection bit: 1:AES; 0:SM4.	0
[7:6]	Reserved	RO	Reserved.	0
[5:4]	MASK_ED_CLK_PRE	RW	Encryption and decryption module clock frequency division factor, Calculation: EDclk=480M/ED_CLK_PRE. The minimum value is 2, and writing 1 is equivalent to turning off ECDC module operation clock.	10b
3	bEDMOD_SELT	RW	1: Decryption mode;	0

			0: Encryption mode.	
2	bRDDAT_ED_EN	RW	Enable to write SD data for encryption and decryption control bit: 1: Encryption and decryption; 0: No action.	0
1	bWRDAT_ED_EN	RW	Enable to read SD data for encryption and decryption control bit: 1: Encryption and decryption; 0: No action.	0
0	bKEYE_EN	RW	Key extension function enable control bit, high level pulse enable.	0

Note: When bKEYE_EN bit is used, it needs to be set high and then low.

AES/SM4 interrupt flag register (AES_SM4_INT_FG)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved.	0
16	bKEYE_ACT_IF	RW1	Key extension completion interrupt flag bit, cleared by writing 1: 1: Key expansion completion trigger; 2. No event.	0
[15:0]	Reserved	RO	Reserved.	0

User key register group (AES_SM4_KEYn) (n=0-7)

Bit	Name	Access	Description	Reset value
[31:0]	AES_SM4_KEY7	RW	User key 223-256 bits.	x
[31:0]	AES_SM4_KEY6	RW	User key 192-223 bits.	x
[31:0]	AES_SM4_KEY5	RW	User key 160-191 bits.	x
[31:0]	AES_SM4_KEY4	RW	User key 128-159 bits.	x
[31:0]	AES_SM4_KEY3	RW	User key 96-127 bits.	x
[31:0]	AES_SM4_KEY2	RW	User key 64-95 bits.	x
[31:0]	AES_SM4_KEY1	RW	User key 32-63 bits.	x
[31:0]	AES_SM4_KEY0	RW	User key 0-31 bits.	x

Count value register group (AES_SM4_IVn) (n=0-3)

Bit	Name	Access	Description	Reset value
[31:0]	AES_SM4_IV3	RW	Count value of 96-127 bits.	x
[31:0]	AES_SM4_IV2	RW	Count value of 64-95 bits.	x
[31:0]	AES_SM4_IV1	RW	Count value of 32-63 bits.	x
[31:0]	AES_SM4_IV0	RW	Count value of 0-31 bits.	x

12.6 Data Storage Encryption and Decryption Applications

12.6.1 Data Encryption Function Configuration

1. Set AES/SM4 control register AES_SM4_CTRL: set bCLR_ALL_IF bit to 1, clear interrupt, select AES or SM4 algorithm, select ECB or CTR mode, and set key length. Note that SM4 algorithm only supports 128-bit key length;
2. Set user key register group and fill in the key. If ETC mode is used, it is also required to set the group value of count value register;

3. Set bKEYE_EN of control register AES_SM4_CTRL, set it to 1 and then to 0, to enable the key expansion;
4. Query the interrupt flag register AES_SM4_INT_FG and wait for the key extension to complete the interrupt. Optionally, turn on the key expansion completion interrupt enable bit bKEYE_ACT_IE of control register and wait for the interrupt to be triggered;
5. Clear the interrupt. Set bEDMOD_SELT bit of the control register to 0, select the encryption mode, and set the bRDDAT_ED_EN bit to 1, to enable the encryption function when transmitting data from SRAM to SD, or set bWRDAT_ED_EN bit to 1, to enable the encryption function when transmitting data from SD to SRAM.

12.6.2 Data Decryption Function Configuration

1. Set AES/SM4 control register AES_SM4_CTRL: set bCLR_ALL_IF bit to 1, clear interrupt, select AES or SM4 algorithm, select ECB or CTR mode, and set key length. Note that SM4 algorithm only supports 128-bit key length;
2. Set user key register group and fill in the key. If CTR mode is used, it is also required to set the group value of count value register;
3. Set bKEYE_EN of control register AES_SM4_CTRL, set it to 1 and then to 0, to enable the key expansion;
4. Query the interrupt flag register AES_SM4_INT_FG and wait for the key extension to complete the interrupt. Optionally, turn on the key expansion completion interrupt enable bit bKEYE_ACT_IE of control register and wait for the interrupt to be triggered;
5. Clear the interrupt. Set bEDMOD_SELT bit of the control register to 1, select the decryption mode, and set the bRDDAT_ED_EN bit to 1, to enable the decryption function when transmitting data from SRAM to SD, or set bWRDAT_ED_EN bit to 1, to enable the decryption function when transmitting data from SD to SRAM.

Chapter 13 SATA Controller

13.1 SATA Controller Introduction

CH568 is embedded with SATA controller, with dual roles of host controller and device controller. The controller will automatically adjust the data stream transmission and extract the complete information frame structure. The user can construct the frame information and start sending according to the ATA protocol, and the controller will automatically add primitives and send them to the physical medium.

The features of SATA controller are as follows:

1) Support SATA Host functions and USB Device functions;
2) Support 1.5G mode and 3G mode;
3) Support flow control and power management;
4) Support data packets up to 2048 double-words, with built-in FIFO, and support interrupt and DMA;
5) Support double buffer mode for receiving/sending data.

13.2 SATA Register Description

SATA related register physical base address: 0x0040 B000

Table 13-1 List of SATA Registers

Name	Offset address	Description	Reset value
SATA_CTRL	00h	SATA control register	8h07
SATA_PM_CTRL	01h	SATA power management register	8h00
SATA_MOD	02h	SATA mode control register	8h00
SATA_INT_EN	04h	SATA interrupt enable register	16h0000
SATA_RX_LEN	08h	SATA receiving length register	16hxxxx
SATA_INT_FG	0Ch	SATA interrupt flag register	16h0000
SATA_INT_ST	0Eh	SATA interrupt status register	8h00
SATA_TX_LEN	10h	SATA sending length register	16hxxxx
SATA_RTX_CTRL	12h	SATA transceiver controller	8h00
SATA_DATA0	14h	SATA data register 0. The first DWORD sent and received is stored in this register, and it is used in conjunction with SATA_DMA0 define buffer.	32h0000 0000
SATA_DATA1	18h	SATA data register 1. The first DWORD sent and received is stored in this register, and it is used in conjunction with SATA_DMA1 define buffer.	32h0000 0000
SATA_DMA0	1Ch	Start address of DMA0 buffer	16hxxxx
SATA_DMA1	1Eh	Start address of DMA1 buffer	16hxxxx

SATA control register (SATA_CTRL)

Bit	Name	Access	Description	Reset value
7	bSC_DMA_EN	RW	DMA enable control bit: 1: Enable DMA function; 0: Disable DMA.	0
6	bSC_CONT_EN	RW	CONTp primitive enable control bit:	0

			1: Enable; 0: Disable.	
5	bSC_HOST_MODE	RW	SATA working mode selection bit: 1: SATA host mode; 0: SATA device mode.	0
4	bSC_FORCE_1P5G	RW	SATA speed type selection bit: 1: Forced to work in 1.5G mode; 0: Normal mode. At this time, the speed depends on bSMS_SPEED_TYPE bit in SATA_MIS_ST register.	0
3	bSC_INT_BUSY	RW	Automatical suspend enable bit before SATA transmission completion interrupt flag is not cleared: 1: Automatically suspend before interrupt flag SIF_TRANSFER is not cleared. For receiving, the R_RDY primitive is automatically not returned. For sending, subsequent transmissions are automatically suspended; 0: Not suspend.	0
2	bSC_RESET_PHY	RW	SATA physical layer software reset control bit: 1: Physical layer reset; 0: The physical layer is working normally. It is ready to send COMRESET in host mode, and ready to send COMINIT in device mode.	1
1	bSC_RESET_LINK	RW	SATA link layer software reset control bit: 1: Link layer reset; 0: The link layer is working normally.	1
0	bSC_CLR_ALL	RW	1: Empty SATA interrupt flag and FIFO, needs software to clear; 0: Not empty.	1

SATA power management register (SATA_PM_CTRL)

Bit	Name	Access	Description	Reset value
7	bSPC_OFFLINE	RW	Reserved.	0
6	bSPC_LISTEN	RW	Reserved.	0
5	bSPC_SLUMBER_S	RW	Reserved.	0
4	bSPC_SLUMBER	RW	1: Enter SLUMBER sleep mode; 0: Normal mode.	0
3	bSPC_PARTIAL	RW	1: Enter PARTIAL power saving mode; 0: Normal mode.	0
2	bSPC_SEND_PMREQ_P	RW	This bit changes from 1 to 0, and then a request is sent to enter PARTIAL power saving mode.	0
1	bSPC_SEND_PMREG_S	RW	This bit changes from 1 to 0, and then a request is sent to enter SLUMBER sleep mode.	0

0	bSPC_DENY	RW	Power management control bit: 1: Not support power management; 0: Support power management.	0

SATA mode control register (SATA_MOD)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved.	0
2	bSM_RX_EN	RW	1: Enable receiving; 0: Disable receiving.	0
1	bSM_TX_EN	RW	1: Enable sending; 1: Disable sending;	0
0	bSM_BUF_MOD	RW	Data buffer mode control bit.	0

The data buffer modes of SATA receiver-transmitter are controlled by a combination of bSM_RX_EN and bSM_TX_EN and bSM_BUF_MOD, refer to the following table for details. In the double-buffer mode, the DMA0 buffer will be selected based on bSRC_*_TOG=0 and DMA1 buffer will be selected based on bSRC_*_TOG=1 during SATA data transmission. bSRC_*_AUTO_TOG can be used to realize automatic switch.

Table 14-2 Receiver/Transmitter Buffer Mode

bSM_RX_EN	bSM_TX_EN	bSM_BUF_MOD	Description: Arrange from low to high with SATA_DMA as start address
0	0	x	The transceiver buffer area is disabled and the SATA_DMA buffer is not used.
1	0	0	Single 2048 double-word receiver buffer, SATA_DMA0 address: receive.
1	0	1	Double 2048 double-word receiver buffers, selected by bSRC_R_TOG. All 4096 double-words are arranged as follows: SATA_DMA0 address: Receive when bSRC_R_TOG=0; SATA_DMA1 address: Receive when bSRC_R_TOG=1.
0	1	0	Single 2048 double-word transmitter buffer, SATA_DMA0 address: Transmit.
0	1	1	Double 2048 double-word transmitter buffer, selected by bSRC_T_TOG. All 4096 double-words are arranged as follows: SATA_DMA0 address: Receive when bSRC_T_TOG=0; SATA_DMA1 address: Receive when bSRC_T_TOG=1.
1	1	x	Single 2048 double-word receiver buffer, single 2048 double-byte transmitter buffer: SATA_DMA0 address: Receive; SATA_DMA1 address: Transmit;

Note: If you need to use double-buffer mode for both receiving and transmitting data, you need to manually switch the receiving and transmission mode configuration.

SATA interrupt enable register (SATA_INT_EN)

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved.	0000b
11	bSIE_COMINIT	RW	COMINIT receive interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
10	bSIE_PHYERR	RW	Physical layer connection error event interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
9	bSIE_PM_REQ	RW	Power management request interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
8	bSIE_PM_RES	RW	Power management request response interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
7	bSIE_TRAN_INT	RW	Transmission suspended (SYNCp received) interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
6	bSIE_HOLD	RW	Data receiving or sending wait (HOLDp received) interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
5	bSIE_DMAT	RW	Terminate DMA data transmission (DMATp received) interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
4	bSIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	bSIE_COLLIDE	RW	Bus conflict: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	bSIE_RECV_OK	RW	Receiving completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	bSIE_TRAN_OK	RW	Transmission completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	bSIE_PHYRDY	RW	Physical layer connection or disconnection event interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

SATA receiving length register (SATA_RX_LEN)

Bit	Name	Access	Description	Reset value
[15:0]	SATA_RX_LEN	RO	Count of current received data, the lowest 2	xxxxh

			bits are fixed to 0, and the highest 2 bits are fixed to 0.	

SATA interrupt flag register (SATA_INT_FG)

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved.	0
12	bSIF_COMINIT	RW1	Receive COMINIT/COMRESET interrupt flag bit, cleared by writing 1: 1: COMINIT/COMRESET trigger is received; 0: No event.	0
11	bSIF_PHYERR	RW1	Physical layer connection error event interrupt flag bit, cleared by writing 1: 1: Physical layer connection error trigger is detected; 0: No event.	0
10	bSIF_PM_PARTIAL	RW1	Request or be requested to enter PARTIAL power saving mode flag bit, cleared by writing 1: 1: After receiving this flag, the power management response will end, and SATA_PM_CTR register needs to be set to enter the corresponding mode; 0: No event.	0
9	bSIF_PM_SLUMBER	RW1	Request or be requested to enter the SLUMBER sleep mode flag bit, cleared by writing 1: 1: After receiving this flag, the power management response will end, and SATA_PM_CTR register needs to be set to enter the corresponding mode; 0: No event.	0
8	bSIF_PM_NAK	RW1	Power management response interrupt flag bit, cleared by writing 1: 1: Power management is not supported; PMNAKp will be received after LINK has sent PMREQ_Sp/PMREQ_Pp, or PMNAKp will be returned after PMREQ_Sp/PMREQ_Pp has been received; 0: No event.	0
7	bSIF_TRAN_INT	RW1	Current transmission suspended interrupt flag bit, cleared by writing 1: 1: SYNCp event trigger is received in the process of sending data; 0: No event.	0
6	bSIF_HOLD	RW1	Data receiving or sending wait (HOLDp received) flag bit, cleared by writing 1: 1: HOLDp event trigger is received;	0

			0: No event.	
5	bSIF_DMAT	RW1	Terminate DMA data transmission interrupt flag bit, cleared by writing 1: 1: DMATp event trigger is received; 2. No event.	0
4	bSIF_FIFO_OV	RW1	FIFO overflow interrupt flag bit, cleared by writing 1: 1: FIFO overflow trigger; 2. No event.	0
3	bSIF_COLLIDE	RW1	In the master mode, bus conflict flag bit, cleared by writing 1: 1: When sending data, bus conflict occurs. 2. No event.	0
2	bSIF_RECV_OK	RW1	Data reception completion flag bit, cleared by writing 1: 1: One frame of data is received and triggered; 2. No event.	0
1	bSIF_TRAN_OK	RW1	Data sending completion flag bit, cleared by writing 1: 1: One frame of data is sent and triggered; 2. No event.	0
0	bSIF_PHYRDY	RW1	Physical layer connection or disconnection event flag bit, cleared by writing 1: 1: Physical layer connection or disconnection event trigger is detected; 2. No event.	0

Note: 1. When bSIF_DMAT interrupt is received, it indicates that some of data being sent has been transmitted, and the data frame received on the other side is complete, and CRC and EOF will still be sent. So bSIF_TRAN_OK interrupt will still be generated.

2. When bSIF_TRAN_OK interrupt is detected, bSIS_CRC_OK=1 and there is no bSIF_DMAT interrupt during this period, the current data has been sent correctly, otherwise, MCU needs to restart to send. If an error occurs during data reception, bSIF_RECV_OK interrupt will not be generated. In addition, if bSRC_R_AUTO_TOG or bSRC_T_AUTO_TOG is 1, any sending error or receiving error will not be automatically reversed.

SATA interrupt status register (SATA_INT_ST)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	
6	bSMS_SPD_TYPE	RO	Current working speed mode: 1: 3G mode; 0: 1.5G mode.	0
5	bSMS_R_FIFO_RDY	RO	SATA receiver FIFO data ready status: 1: Receiver FIFO is non-empty; 0: The receiver FIFO is empty.	0
4	bSIS_LINK_FREE	RO	SATA link layer idle status bit: 1: Idle, data can be sent; 0: Busy.	0
3	bSIS_HOLD	RO	1: It is in HOLD status currently, and the	0

			current transmission can be interrupted by bSRC_SYNC_ESCAPE; 0: Not in HOLD status.	
2	bSIS_RECV_CRC_OK	RO	Receiving data frame check status bit, check this bit after bSIF_RECV_OK interrupt: 1: If it is checked as correct, the physical layer will return R_OK; 1: If it is checked as error, the physical layer will return R_ERR;	0
1	bSIS_TRAN_CRC_OK	RO	Transmitting data frame check status bit, check this bit after bSIF_TRAN_OK interrupt: 1: If transmission succeeds, it will be received by the other party; 0: If transmission fails, it needs to be resent by MCU.	0
0	bSIS_PHYRDY	RO	PHY_READY status bit: 1: Physical layer is connected normally; 0: Physical layer is disconnected.	0

SATA sending length register (SATA_TX_LEN)

Bit	Name	Access	Description	Reset value
[15:0]	SATA_TX_LEN	RW	Number of data bytes sent, the lowest 2 bits are fixed to 0 (4 bytes are aligned), and the highest 2 bits are fixed to 0.	xxxxh

SATA transceiver control register (SATA_RTX_CTRL)

Bit	Name	Access	Description	Reset value
7	bSRC_R_AUTO_TOG	RW	In the receiving double buffer mode, the buffer is automatically switched: 1: Enable; 0: Disable.	0
6	bSRC_T_AUTO_TOG	RW	In the sending double buffer mode, the buffer is automatically switched: 1: Enable; 0: Disable.	0
5	bSRC_R_TOG	RW	In the receiving double buffer mode, data storage position: 1: The received data is stored in SATA_DMA0 buffer; 0: The received data is stored in SATA_DMA1 buffer.	0
4	bSRC_T_TOG	RW	In the transmitting double buffer mode, data sending position: 1: The transmitted data is stored in SATA_DMA0 buffer; 0: The transmitted data is stored in SATA_DMA1 buffer.	0

3	Reserved	RO	Reserved.	0
2	bSRC_R_READY	RW	1: Ready to receive data; 0: Prohibit receiving data.	0
1	bSRC_T_READY	RW	1: Ready to send data, start sending, and needs to be manually cleared after sending; 0: Not send.	0
0	bSRC_SYNC_ESCAPE	RW	1: Force to suspend the current transmission and send SYNCp primitive; 0: No action.	0

SATA data buffer 0 (SATA_DATA0):

Bit	Name	Access	Description	Reset value
[31:0]	SATA_DATA0	RW	SATA data register 0. The first double-word data sent and received by LINK is stored in this register.	0

SATA data buffer 1 (SATA_DATA1):

Bit	Name	Access	Description	Reset value
[31:0]	SATA_DATA1	RW	SATA data register 1. The first double-word data sent and received by LINK is stored in this register.	0

Start address of SATA buffer 0 (SATA_DMA0)

Bit	Name	Access	Description	Reset value
[15:0]	SATA_DMA0	RW	Start address of buffer 0, the lowest 3 bits are fixed to 0 (8 bytes are aligned), and the highest 1 bit is fixed to 0.	xxxxh

Start address of SATA buffer 1 (SATA_DMA1)

Bit	Name	Access	Description	Reset value
[15:0]	SATA_DMA1	RW	Start address of buffer 1, the lowest 3 bits are fixed to 0 (8 bytes are aligned), and the highest 1 bit is fixed to 0.	xxxxh

13.3 Programming Guide

13.3.1 SATA Device/Host Connection

The SATA controller can automatically establish the communication link between device and host, and the user only needs to query the corresponding flag bit. This greatly simplifies the control of underlying communication timing for users.

Master mode configuration:

1. Initialize the host function: set bSC_HOST_MODE, bSC_DMA_EN and bSC_INT_BUSY bits of SATA_CTRL register to 1;
2. Set bSC_RESET_PHY, bSC_RESET_LINK and bSC_CLR_ALL bits of SATA_CTRL register to 0;
3. Set the receiving and transmission mode register SATA_MOD and buffer register DMA address SATA_DMA0 and SATA_DMA1;
4. Optionally, turn on the power management function;

5. Set bSRC_R_READY bit of SATA_RTX_CTRL register to 1, and enable data reception;
6. Waiting for device connection: If the query flag register bSIF_PHYRDY is set, you need to query the bSIS_PHYRDY bit of SATA_INT_ST register again to detect the device connection. If bSIF_PHYERR is set, there is an error in the process of establishing communication link, so you need to reconnect, clear the interrupt flag, set bSC_RESET_PHY bit to high and then low, and re-establish communication link with the device.

Device mode configuration:
1. Initialize the device function: set bSC_DMA_EN and bSC_INT_BUSY bits of SATA_CTRL register to 1;
2. Set bSC_RESET_PHY, bSC_RESET_LINK and bSC_CLR_ALL bits of SATA_CTRL register to 0;
3. Set the receiving and transmission mode register SATA_MOD and buffer register DMA address SATA_DMA0 and SATA_DMA1;
4. Optionally, turn on the power management function;
5. Set the bSRC_R_READY bit of SATA_RTX_CTRL register to 1, and enable data reception;
6. Wait for the bSIF_PHYRDY bit of interrupt flag register to be set. At this time, the communication link is established and the device can send a device status frame (FIS=34h) according to the ATA protocol.

13.3.2 Data Transmission

1. Construct FIS frame according to ATA protocol, and fill the first 4 bytes to SATA_DATAx register;
2. Write the first address of subsequent frame content to SATA_DMAx register. 8 bytes must be aligned for this address;
3. Fill the number of frame length bytes to SATA_TX_LEN register, and the number of bytes must be a multiple of 4;
4. Query the bSIS_LINK_FREE bit of SATA_INT_ST register. When this bit is 1, set bSRC_T_READY bit to start sending;
5. Wait for bSIF_TRAN_OK bit of interrupt flag register and bSIS_TRAN_CRC_OK bit of status register to 1, which means that the data has been correctly sent to the receiving end, and the bSRC_T_READY bit and flag need to be cleared. Otherwise, the frame needs to be resent.

13.3.3 Data Reception

1. Set the receiving data address of SATA_DMAx register (8 bytes are aligned), and set the bSRC_R_READY bit of SATA_RTX_CTRL register;
2. Wait for the bSIF_RECV_OK bit of the interrupt flag register and the bSIS_RECV_CRC_OK bit of the status register to 1, which means that the data is received correctly and the interrupt flag is cleared. Otherwise, it is required to continue to wait for reception, and the underlying hardware will also send the receiving error (R_ERRp).
3. Get the received frame from the address buffer area defined by SATA_DMAx register and SATA_DATAx, and read the total number of bytes received from the SATA_RX_LEN register.

Chapter 14 Interrupt

14.1 Interrupt Controller

CH568 supports multiple interrupt sources, with a total of 16 peripheral interrupt sources, including UART, SPI, TMR, USB, GPIO, etc.

Table 14-1 Interrupt Vector

Default priority sequence	Interrupt No.	Priority	Name	Description
High priority ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ Low priority	0	Level 2, which can be set	SOFT	Software interrupt
	1	Level 2, which can be set	TMR0	Timer 0 interrupt
	2	Level 2, which can be set	GPIO	GPIO port interrupt
	3	Level 2, which can be set	SPI0	SPI0 interrupt
	4	Level 2, which can be set	SATA	SATA interrupt
	5	Level 2, which can be set	TMR1	Timer1 interrupt
	6	Level 2, which can be set	TMR2	Timer2 interrupt
	7	Level 2, which can be set	UART0	UART0 interrupt
	8	Level 2, which can be set	USB1	USB1 interrupt
	9	Level 2, which can be set	SDC	SD controller interrupt
	10	Level 2, which can be set	ECDC	Encryption and decryption controller interrupt
	11	Level 2, which can be set	LED	LED controller interrupt
	12	Level 2, which can be set	SPI1	SPI1 interrupt
	13	Level 2, which can be set	UART1	UART1 interrupt
	14	Level 2, which can be set	UART2	UART2 interrupt
	15	Level 2, which can be set	UART3	UART3 interrupt

14.2 Interrupt Instruction

1) To use the interrupt function of CH568 chip, you need to add the following functions to the software code to enable the system interrupt control.
void Interrupt_init(void);
2) Enable the corresponding interrupt control bit of interrupt enable register of corresponding peripheral module, to trigger the interrupt under the corresponding trigger condition.
3) Refer to the following example for the writing of interrupt function. Taking TMR0 as an example, the interrupt number is 1:

```
__attribute__( ( interrupt ( "id=1" ) ) )    void TIME0_Deal()
{
    static UINT16 j=0;
    if(R8_TMR0_INT_FLAG & RB_TMR_IF_CYC_END)
    {
        j++;
        if(j>300){
            printf("*\n");
            j = 0;
        }
```

```
            R8_TMR0_INT_FLAG |= RB_TMR_IF_CYC_END;          //Clear flag
        }
    }
```

Chapter 15 Parameters

15.1 Absolute Maximum Value

Critical value or exceeding the absolute maximum value may cause the chip to work abnormally or even be damaged.

Table 15-1 Absolute Maximum Value Parameters

Name	Parameter description		Min.	Max.	Unit
TA	Ambient temperature during operation	VCC33=3.3V V33IO1/2/3=3.3V VCC12A=1.2V	-40	85	℃
TS	Ambient temperature during storage		-55	125	℃
VCC33	System power voltage (VCC33 connects to power, GND to ground)		-0.4	4.2	V
V33IO1	Peripheral group 1 power voltage (V33IO1 connects to power, GND to ground)		-0.4	4.2	V
V33IO2	Peripheral group 2 power voltage (V33IO2 connects to power, GND to ground)		-0.4	4.2	V
V33IO3	Peripheral group 3 power voltage (V33IO3 connects to power, GND to ground)		-0.4	4.2	V
VIO0	Voltage on input or output pin of VCC33 power domain		-0.4	VCC33+0.4	V
VIO1	Voltage on input or output pin of V33IO1 power domain		-0.4	V33IO1+0.4	V
VIO2	Voltage on input or output pin of V33IO2 power domain		-0.4	V33IO2+0.4	V
VIO3	Voltage on input or output pin of V33IO3 power domain		-0.4	V33IO3+0.4	V
VCC12A	SATA-PHY power voltage		-0.3	1.5	V
VIOSATA	Voltage on the SATA-PHY signal pin		-0.3	VCC12A+0.3	V

15.2 Electrical Parameters

Test conditions: TA=25℃, VCC33=3.3V, V33IO1/2/3=3.3V, VCC12A=1.2V, Fsys=96MHz.

Table 15-2 Electrical Parameters

Name	Parameter description		Min.	Typ.	Max.	Unit
VCC33	System supply voltage	VCC33	2.7	3.3	3.6	V
V33IO1	Peripheral group 1 power voltage	V33IO1	1.6	3.3	3.6	V
V33IO2	Peripheral group 2 power voltage	V33IO2	1.6	3.3	3.6	V
V33IO3	Peripheral group 3 power voltage	V33IO3	3.0	3.3	3.6	V
VCC12A	SATA-PHY power voltage	VCC12A	1.15	1.2	1.3	V
ICC	Total supply current during operation		20	45	150	mA
ISLP	Supply current at low-power consumption status I/O pin output with no-load or input with pull-down		240	280	350	uA
VIL	Low-level input voltage (V33IO=3.3V)		-0.4	-	0.7	V
VIH	High-level input voltage (V33IO=3.3V)		2.0	-	V33IO+0.4	V

VIL18	Low-level input voltage (V33IO1/2=1.8V)	-0.4	-	0.5	V
VIH18	High-level input voltage (V33IO1/2=1.8V)	1.2	-	V33IO+0.4	V
VOL	Low level output voltage (6mA draw current)	-	-	0.4	V
VOH	High level output voltage (5mA output current)	V33IO-0.4	-	-	V
IUP	Input current at the input terminal with built-in pull-up resistor	25	45	80	uA
IDN	Input current at the input terminal with built-in pull-down resistor	-25	-45	-80	uA
Vpot	Voltage threshold for VCC12 core power power-on reset	0.6	0.7	0.8	V

15.3 Static Current of Function Module

Test conditions: TA=25°C, VCC33=3.3V, V33IO1/2/3=3.3V, VCC12A=1.2V.

Table 15-3 Dynamic current of Function Module

Frequency / Function Module	30M	60M	96M	120M	Unit
USB	0.2	0.4	0.65	0.79	mA
USB-Phy	10				mA
SATA	0.23	0.48	0.72	0.92	mA
SATA-Phy	50				mA
SDC-96MHz	12.45	13.53	14.84	15.65	mA
SDC-48MHz	6.28	7.39	8.64	9.49	mA
SDC-24MHz	4.16	5.22	6.5	7.32	mA
ECDC-240MHz	15.64	16.88	17.72	19.3	mA
ECDC-160MHz	10.99	12.25	13.79	14.69	mA
TMR+UART+SPI+PWM	0.17	0.33	0.51	0.7	mA
PLL	7				mA
Core+BUS+DMA	24	27	31	34	mA

15.4 Timing Parameters

Test conditions: TA=25°C, VCC33=3.3V, V33IO1/2/3=3.3V, Fsys=96MHz.

Table 15-4 Timing Sequence Parameters

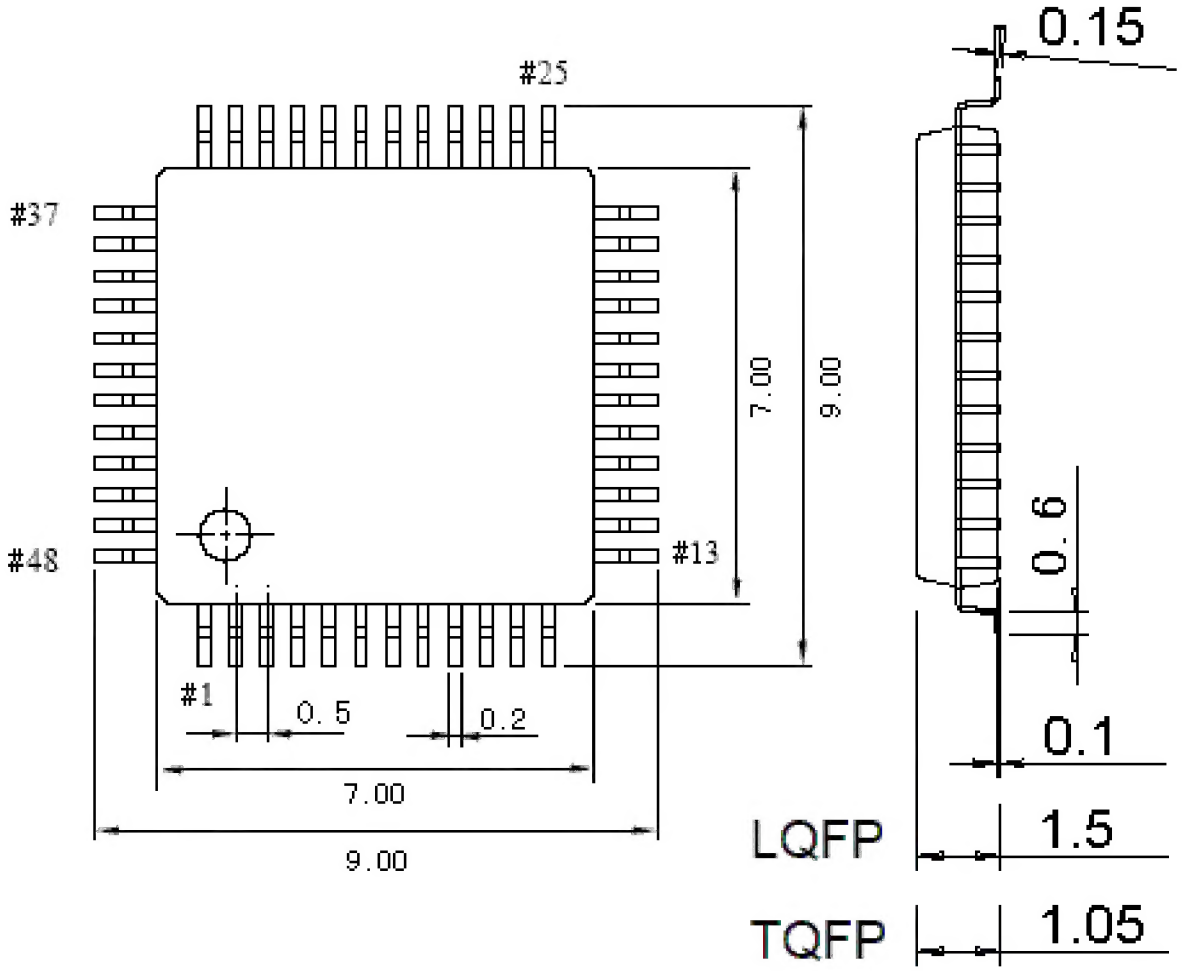
Name	Parameter description	Min.	Typ.	Max.	Unit
Trst	Valid signal width of external reset input RST#	50	2*Tsys	-	ns
Tpro	Reset delay after power-on reset	22	32	50	mS
Tsro	Reset delay after external/software reset input + load time	8	8.8	10	mS
TWAK	Wake-up time when exiting from low-power status	0.2	1	5	mS

Chapter 16 Package

Chip package

Package	Width Of Plastic	Pitch Of Pin		Instruction Of Package	Ordering Information
LQFP-48	7*7mm	0.5mm	19.7mil	Standard LQFP48 pin patch	CH568L

Remarks:
The unit of dimension is mm (millimeters)
The pin center spacing is the nominal value, there is no error, and other dimension error is not greater than ±0.2mm.



Chapter 17 Modification Record

Version	Date	Description
V1.0	November 28, 2016	First release
V1.1	October 29, 2018	Some error descriptions are modified
V1.2	August 23, 2021	Figure of pin arrangement in section 1.1 is updated. Add description and note about SDIO clock pin in section 1.2. Add description of block transmission clock count bit bTM_GAP_STOP in section 12.2.